1.Relational Plot -to see the statistical relation b/w 2 or more variables. -Bivariate analysis
Scater plot and Line plot

In [1]:
```
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.express as px
```

In [2]:
```
tips=sns.load_dataset("tips")
```

In [3]:
```
tips
```

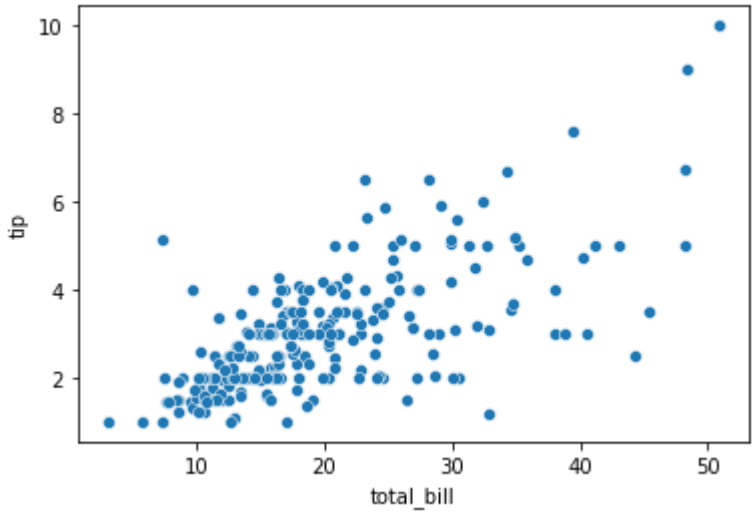
Out[3]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
...	...	...	...	...	...	...	...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	2.00	Male	Yes	Sat	Dinner	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

244 rows × 7 columns

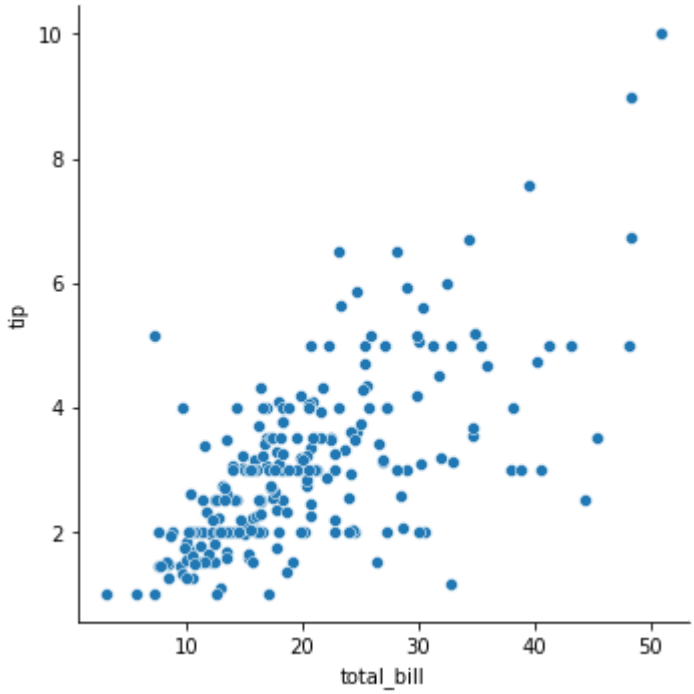
In [4]:
```
#scatter plot->axes level function
sns.scatterplot(data=tips,x="total_bill",y="tip")
```

Out[4]:
```
<AxesSubplot:xlabel='total_bill', ylabel='tip'>
```



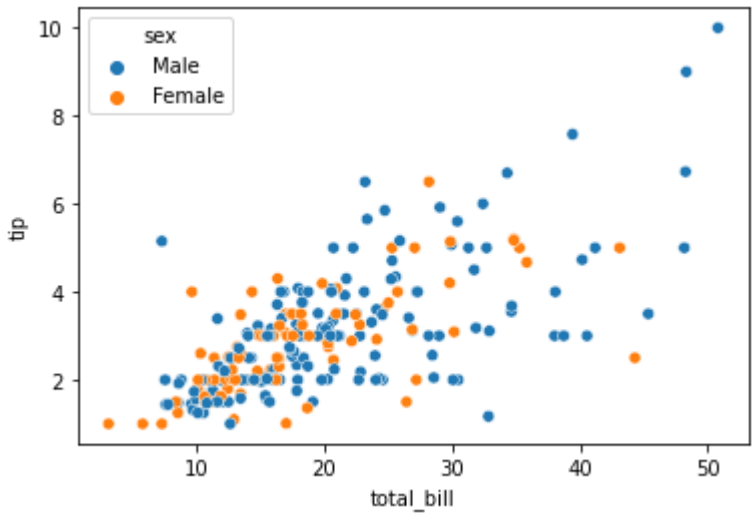
In [5]:
```
sns.relplot(data=tips,x="total_bill",y="tip",kind="scatter")
#relplot is figure level->square shape
```

Out[5]: <seaborn.axisgrid.FacetGrid at 0x1a0e1302760>



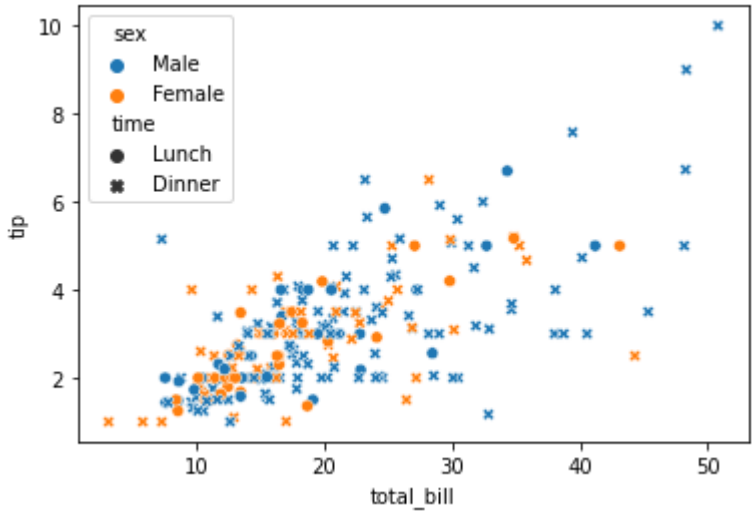
In [6]:
```
sns.scatterplot(data=tips,x="total_bill",y="tip",hue="sex")
```

Out[6]: <AxesSubplot:xlabel='total_bill', ylabel='tip'>



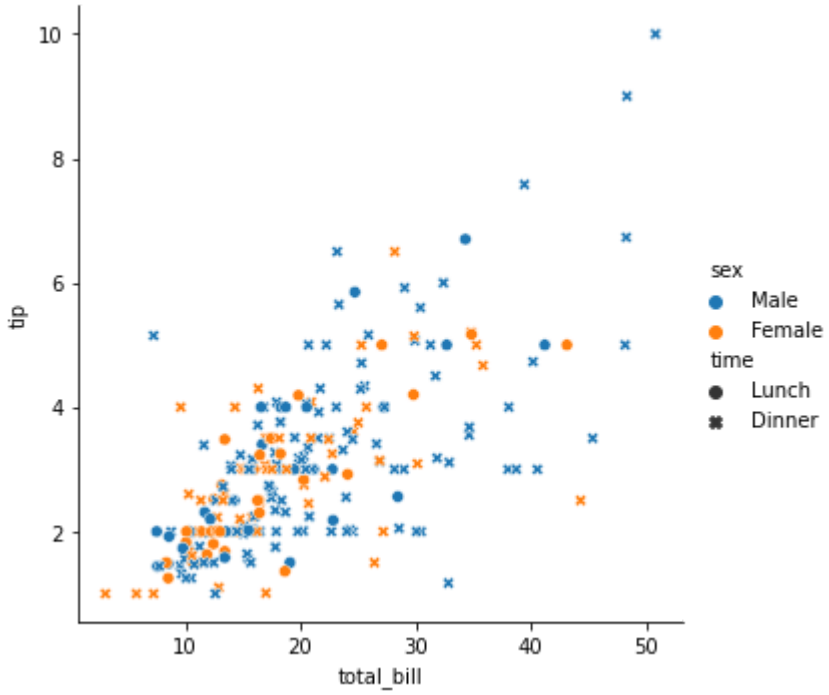
In [7]:
```
sns.scatterplot(data=tips,x="total_bill",y="tip",hue="sex",style="time")
```

Out[7]: <AxesSubplot:xlabel='total_bill', ylabel='tip'>



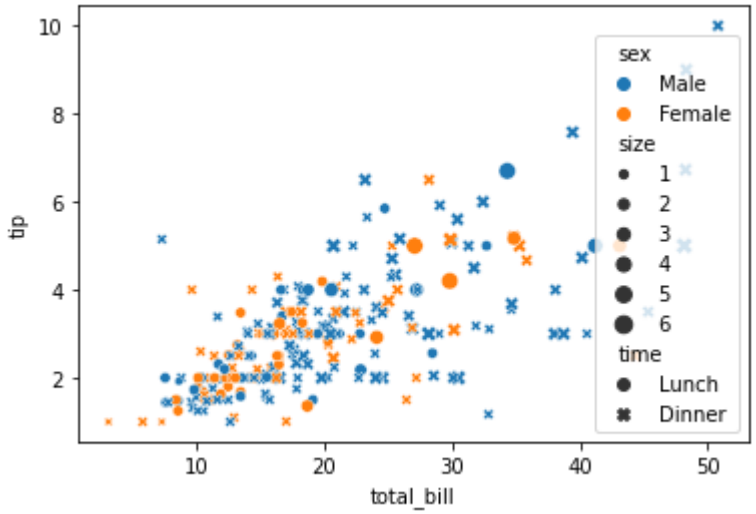
In [8]:
```
sns.relplot(data=tips,x="total_bill",y="tip",kind="scatter",hue="sex",style="time")
```

Out[8]:
```
<seaborn.axisgrid.FacetGrid at 0x1a0e146bf10>
```



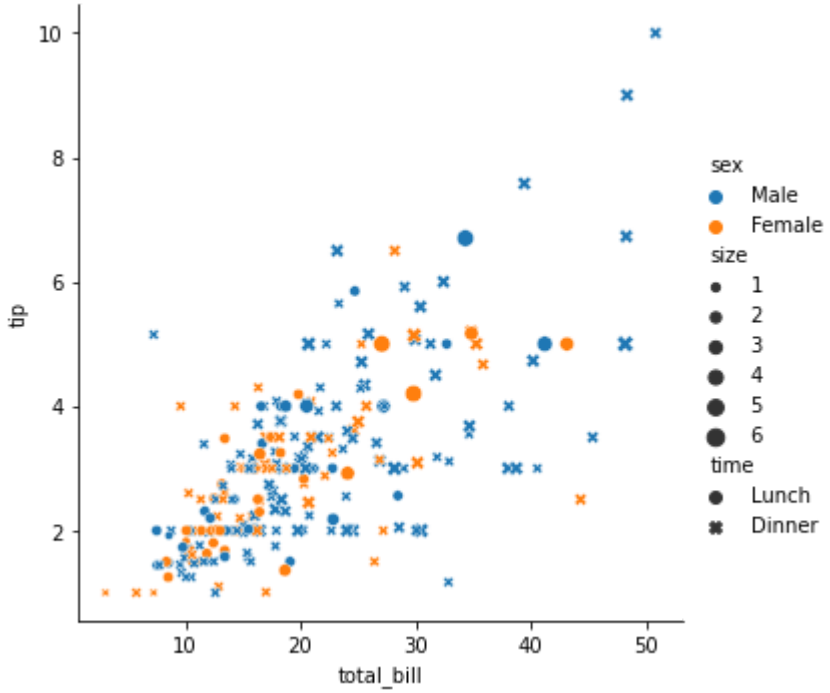
In [9]:
```
sns.scatterplot(data=tips,x="total_bill",y="tip",hue="sex",style="time",size="size"
```

Out[9]:
```
<AxesSubplot:xlabel='total_bill', ylabel='tip'>
```



In [10]:
```
sns.relplot(data=tips,x="total_bill",y="tip",kind="scatter",hue="sex",style="time",
```

Out[10]:
```
<seaborn.axisgrid.FacetGrid at 0x1a0e13e2e50>
```



In [11]:
```
#Line plot
gap=px.data.gapminder()
gap
```

Out[11]:

	country	continent	year	lifeExp	pop	gdpPercap	iso_alpha	iso_num
0	Afghanistan	Asia	1952	28.801	8425333	779.445314	AFG	4
1	Afghanistan	Asia	1957	30.332	9240934	820.853030	AFG	4
2	Afghanistan	Asia	1962	31.997	10267083	853.100710	AFG	4
3	Afghanistan	Asia	1967	34.020	11537966	836.197138	AFG	4
4	Afghanistan	Asia	1972	36.088	13079460	739.981106	AFG	4
...	...	...	...	...	...	...	...	...
1699	Zimbabwe	Africa	1987	62.351	9216418	706.157306	ZWE	716
1700	Zimbabwe	Africa	1992	60.377	10704340	693.420786	ZWE	716
1701	Zimbabwe	Africa	1997	46.809	11404948	792.449960	ZWE	716
1702	Zimbabwe	Africa	2002	39.989	11926563	672.038623	ZWE	716
1703	Zimbabwe	Africa	2007	43.487	12311143	469.709298	ZWE	716

1704 rows × 8 columns

In [ ]:
```

```

In [ ]:
```

```

In [12]:
```
temp_df=gap[gap["country"]=="India"]
```

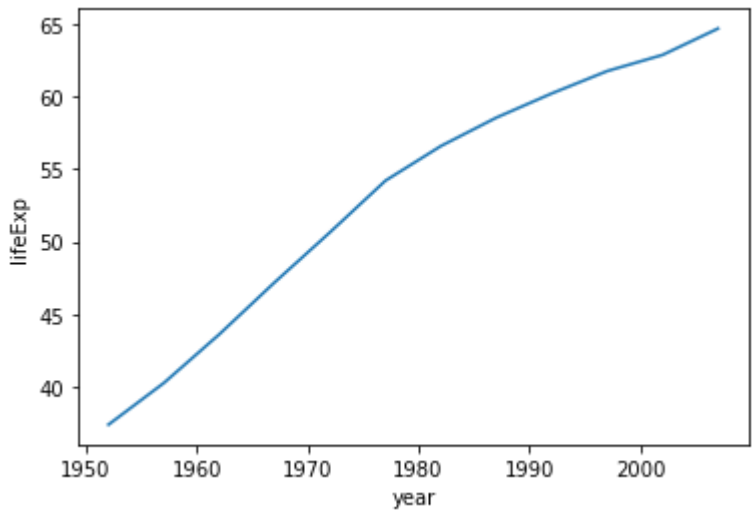
In [13]:
```
temp_df
```

Out[13]:

	country	continent	year	lifeExp	pop	gdpPercap	iso_alpha	iso_num
696	India	Asia	1952	37.373	372000000	546.565749	IND	356
697	India	Asia	1957	40.249	409000000	590.061996	IND	356
698	India	Asia	1962	43.605	454000000	658.347151	IND	356
699	India	Asia	1967	47.193	506000000	700.770611	IND	356
700	India	Asia	1972	50.651	567000000	724.032527	IND	356
701	India	Asia	1977	54.208	634000000	813.337323	IND	356
702	India	Asia	1982	56.596	708000000	855.723538	IND	356
703	India	Asia	1987	58.553	788000000	976.512676	IND	356
704	India	Asia	1992	60.223	872000000	1164.406809	IND	356
705	India	Asia	1997	61.765	959000000	1458.817442	IND	356
706	India	Asia	2002	62.879	1034172547	1746.769454	IND	356
707	India	Asia	2007	64.698	1110396331	2452.210407	IND	356

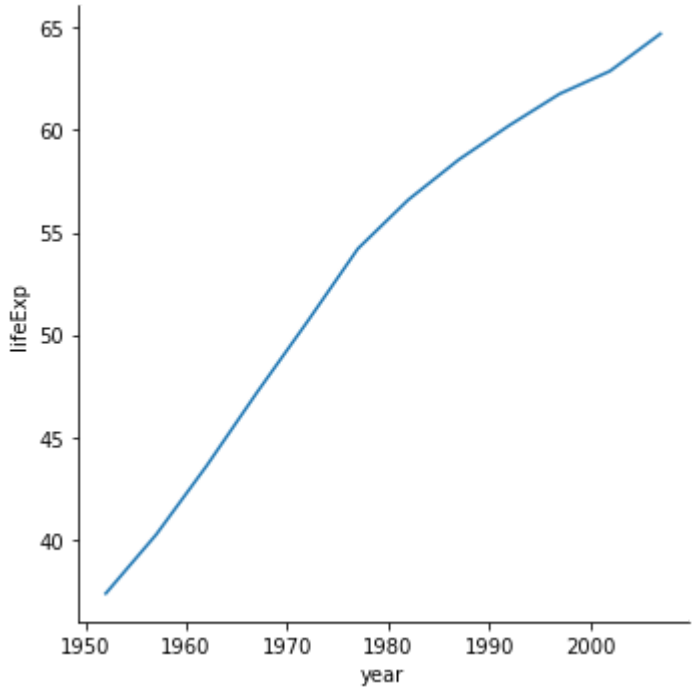
In [14]:
```
#axes level function
sns.lineplot(data=temp_df,x="year",y="lifeExp")
```

Out[14]:
```
<AxesSubplot:xlabel='year', ylabel='lifeExp'>
```



In [15]:
```
#using relplot
sns.relplot(data=temp_df,x="year",y="lifeExp",kind="line")
```

Out[15]:
```
<seaborn.axisgrid.FacetGrid at 0x1a0e16b0f10>
```



In [16]:
```
#hue->style
temp_df1=gap[gap["country"].isin(["India","Pakistan","China"])]
```

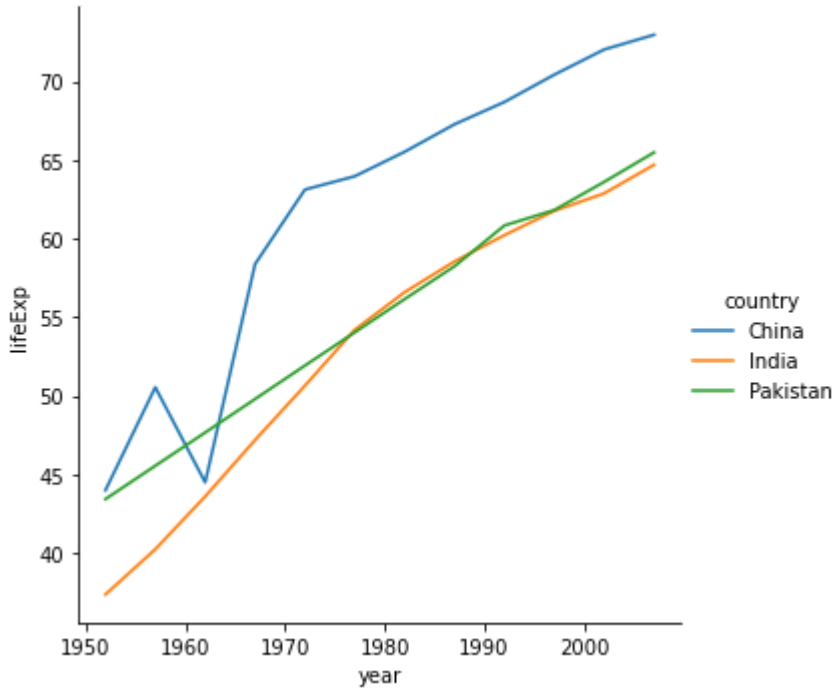
In [17]:
```
temp_df1
```

Out[17]:

	country	continent	year	lifeExp	pop	gdpPercap	iso_alpha	iso_num
288	China	Asia	1952	44.00000	556263527	400.448611	CHN	156
289	China	Asia	1957	50.54896	637408000	575.987001	CHN	156
290	China	Asia	1962	44.50136	665770000	487.674018	CHN	156
291	China	Asia	1967	58.38112	754550000	612.705693	CHN	156
292	China	Asia	1972	63.11888	862030000	676.900092	CHN	156
293	China	Asia	1977	63.96736	943455000	741.237470	CHN	156
294	China	Asia	1982	65.52500	1000281000	962.421381	CHN	156
295	China	Asia	1987	67.27400	1084035000	1378.904018	CHN	156
296	China	Asia	1992	68.69000	1164970000	1655.784158	CHN	156
297	China	Asia	1997	70.42600	1230075000	2289.234136	CHN	156
298	China	Asia	2002	72.02800	1280400000	3119.280896	CHN	156
299	China	Asia	2007	72.96100	1318683096	4959.114854	CHN	156
696	India	Asia	1952	37.37300	372000000	546.565749	IND	356
697	India	Asia	1957	40.24900	409000000	590.061996	IND	356
698	India	Asia	1962	43.60500	454000000	658.347151	IND	356
699	India	Asia	1967	47.19300	506000000	700.770611	IND	356
700	India	Asia	1972	50.65100	567000000	724.032527	IND	356
701	India	Asia	1977	54.20800	634000000	813.337323	IND	356
702	India	Asia	1982	56.59600	708000000	855.723538	IND	356
703	India	Asia	1987	58.55300	788000000	976.512676	IND	356
704	India	Asia	1992	60.22300	872000000	1164.406809	IND	356
705	India	Asia	1997	61.76500	959000000	1458.817442	IND	356
706	India	Asia	2002	62.87900	1034172547	1746.769454	IND	356
707	India	Asia	2007	64.69800	1110396331	2452.210407	IND	356
1164	Pakistan	Asia	1952	43.43600	41346560	684.597144	PAK	586
1165	Pakistan	Asia	1957	45.55700	46679944	747.083529	PAK	586
1166	Pakistan	Asia	1962	47.67000	53100671	803.342742	PAK	586
1167	Pakistan	Asia	1967	49.80000	60641899	942.408259	PAK	586
1168	Pakistan	Asia	1972	51.92900	69325921	1049.938981	PAK	586
1169	Pakistan	Asia	1977	54.04300	78152686	1175.921193	PAK	586
1170	Pakistan	Asia	1982	56.15800	91462088	1443.429832	PAK	586
1171	Pakistan	Asia	1987	58.24500	105186881	1704.686583	PAK	586
1172	Pakistan	Asia	1992	60.83800	120065004	1971.829464	PAK	586
1173	Pakistan	Asia	1997	61.81800	135564834	2049.350521	PAK	586
1174	Pakistan	Asia	2002	63.61000	153403524	2092.712441	PAK	586
1175	Pakistan	Asia	2007	65.48300	169270617	2605.947580	PAK	586

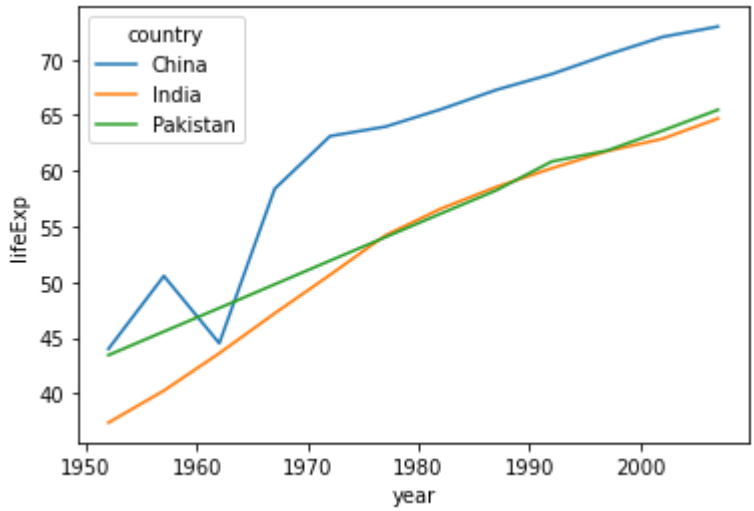
In [18]:
```
sns.relplot(kind="line",data=temp_df1,x="year",y="lifeExp",hue="country")
```

Out[18]:
```
<seaborn.axisgrid.FacetGrid at 0x1a0e1801fa0>
```



In [19]:
```
sns.lineplot(data=temp_df1,x="year",y="lifeExp",hue="country")
```

Out[19]:
```
<AxesSubplot:xlabel='year', ylabel='lifeExp'>
```



In [20]:
```
temp_df1=gap[gap["country"].isin(["India","Brazil","Germany"])]
```

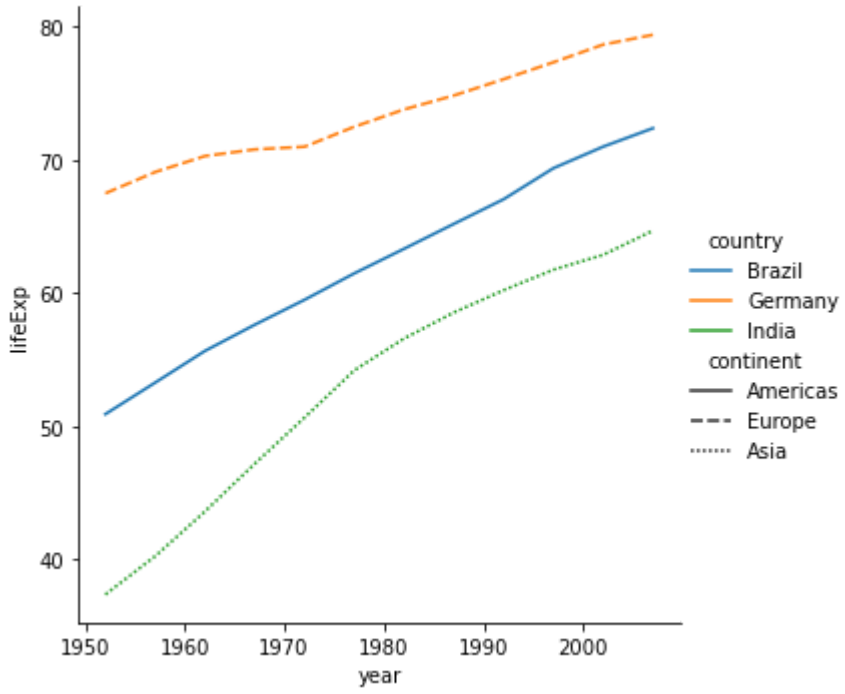
In [21]:
```
temp_df1
```

Out[21]:

	country	continent	year	lifeExp	pop	gdpPercap	iso_alpha	iso_num
168	Brazil	Americas	1952	50.917	56602560	2108.944355	BRA	76
169	Brazil	Americas	1957	53.285	65551171	2487.365989	BRA	76
170	Brazil	Americas	1962	55.665	76039390	3336.585802	BRA	76
171	Brazil	Americas	1967	57.632	88049823	3429.864357	BRA	76
172	Brazil	Americas	1972	59.504	100840058	4985.711467	BRA	76
173	Brazil	Americas	1977	61.489	114313951	6660.118654	BRA	76
174	Brazil	Americas	1982	63.336	128962939	7030.835878	BRA	76
175	Brazil	Americas	1987	65.205	142938076	7807.095818	BRA	76
176	Brazil	Americas	1992	67.057	155975974	6950.283021	BRA	76
177	Brazil	Americas	1997	69.388	168546719	7957.980824	BRA	76
178	Brazil	Americas	2002	71.006	179914212	8131.212843	BRA	76
179	Brazil	Americas	2007	72.390	190010647	9065.800825	BRA	76
564	Germany	Europe	1952	67.500	69145952	7144.114393	DEU	276
565	Germany	Europe	1957	69.100	71019069	10187.826650	DEU	276
566	Germany	Europe	1962	70.300	73739117	12902.462910	DEU	276
567	Germany	Europe	1967	70.800	76368453	14745.625610	DEU	276
568	Germany	Europe	1972	71.000	78717088	18016.180270	DEU	276
569	Germany	Europe	1977	72.500	78160773	20512.921230	DEU	276
570	Germany	Europe	1982	73.800	78335266	22031.532740	DEU	276
571	Germany	Europe	1987	74.847	77718298	24639.185660	DEU	276
572	Germany	Europe	1992	76.070	80597764	26505.303170	DEU	276
573	Germany	Europe	1997	77.340	82011073	27788.884160	DEU	276
574	Germany	Europe	2002	78.670	82350671	30035.801980	DEU	276
575	Germany	Europe	2007	79.406	82400996	32170.374420	DEU	276
696	India	Asia	1952	37.373	372000000	546.565749	IND	356
697	India	Asia	1957	40.249	409000000	590.061996	IND	356
698	India	Asia	1962	43.605	454000000	658.347151	IND	356
699	India	Asia	1967	47.193	506000000	700.770611	IND	356
700	India	Asia	1972	50.651	567000000	724.032527	IND	356
701	India	Asia	1977	54.208	634000000	813.337323	IND	356
702	India	Asia	1982	56.596	708000000	855.723538	IND	356
703	India	Asia	1987	58.553	788000000	976.512676	IND	356
704	India	Asia	1992	60.223	872000000	1164.406809	IND	356
705	India	Asia	1997	61.765	959000000	1458.817442	IND	356
706	India	Asia	2002	62.879	1034172547	1746.769454	IND	356
707	India	Asia	2007	64.698	1110396331	2452.210407	IND	356

In [22]:
```
sns.relplot(kind="line",data=temp_df1,x="year",y="lifeExp",hue="country",style="cor
```

Out[22]:
```
<seaborn.axisgrid.FacetGrid at 0x1a0e1923e50>
```



In [23]:
```
#facet plot->figure function level->work with relplot
#it will not work with scatter and lineplot
tips
```

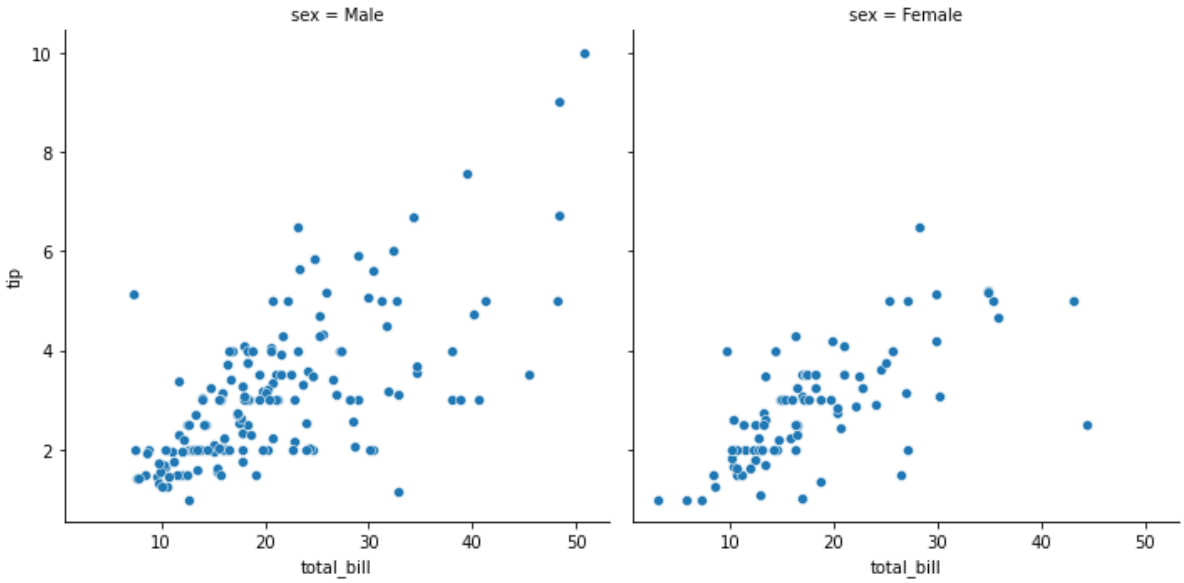
Out[23]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
...	...	...	...	...	...	...	...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	2.00	Male	Yes	Sat	Dinner	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

244 rows × 7 columns

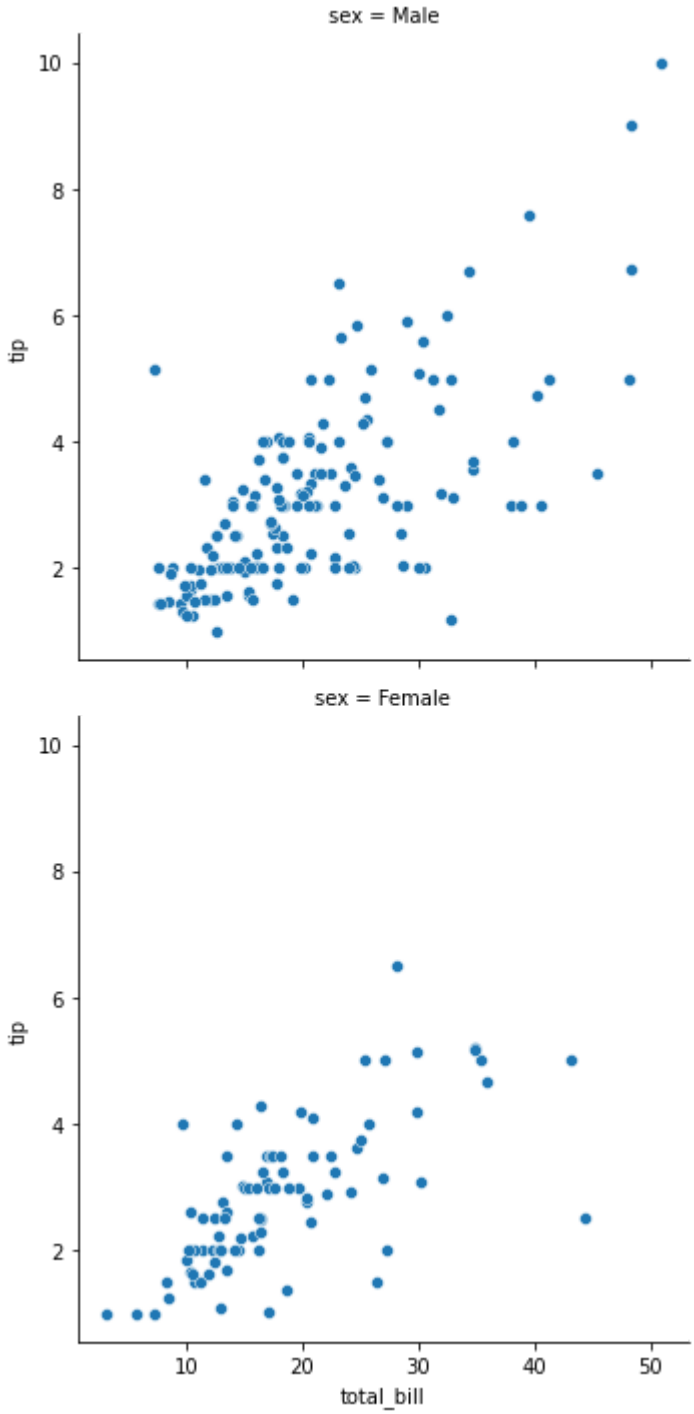
In [24]:
```
sns.relplot(data=tips,x="total_bill",y="tip",kind="scatter",col="sex")
```

Out[24]:
```
<seaborn.axisgrid.FacetGrid at 0x1a0e1932bb0>
```



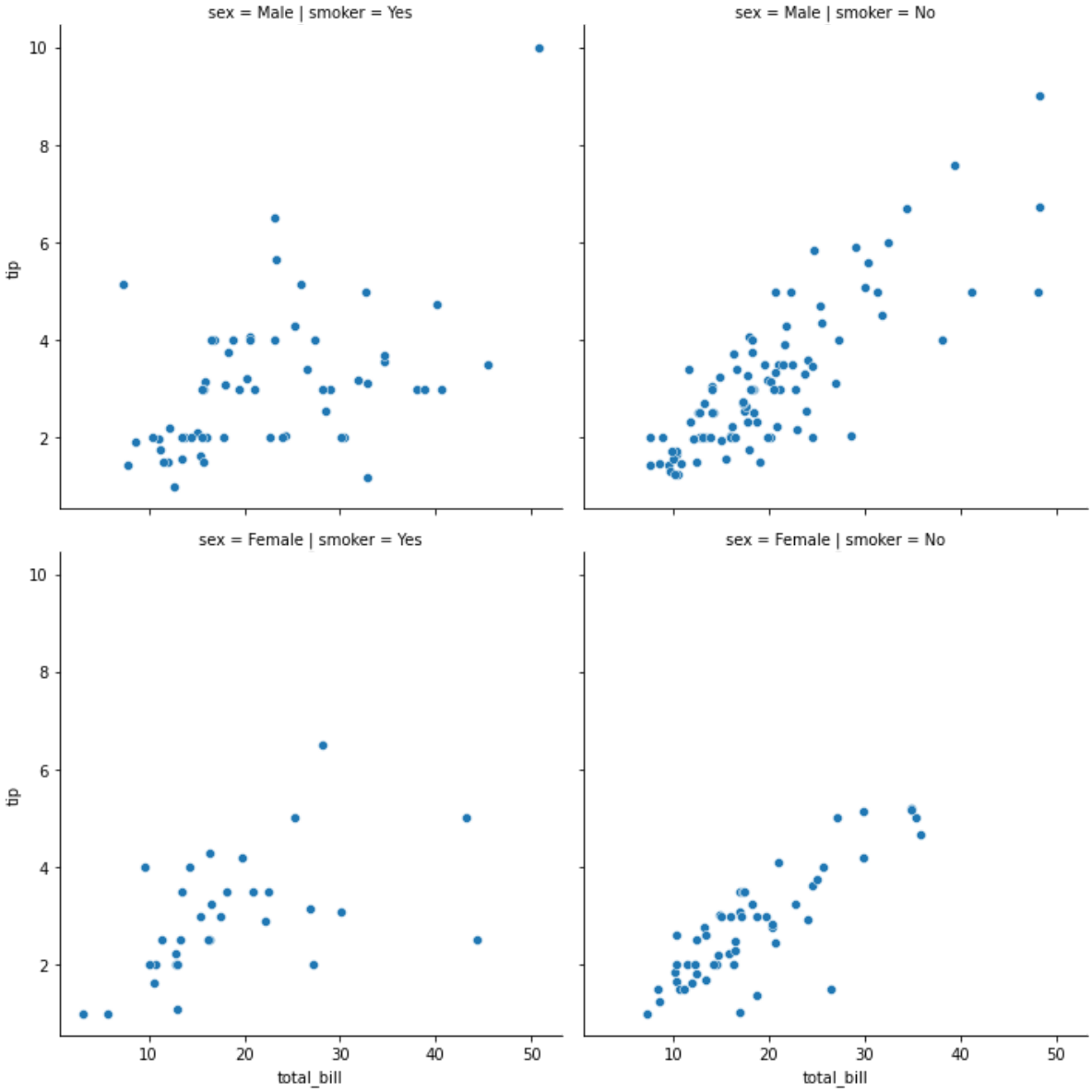
In [25]:
```
sns.relplot(data=tips,x="total_bill",y="tip",kind="scatter",row="sex")
```

Out[25]:
```
<seaborn.axisgrid.FacetGrid at 0x1a0e2a070a0>
```



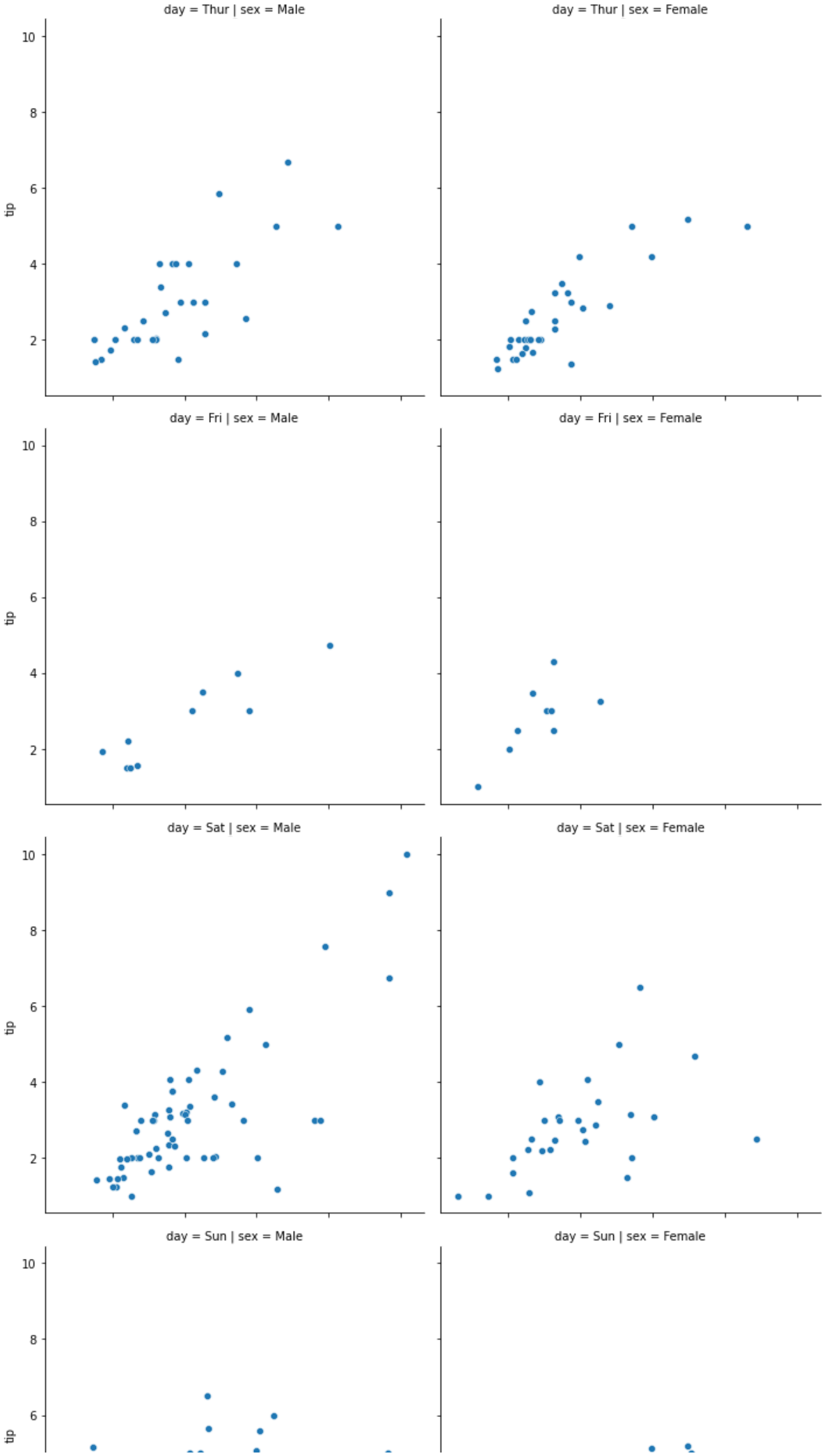
In [26]:
```
sns.relplot(data=tips,x="total_bill",y="tip",kind="scatter",col="smoker",row="sex")
```

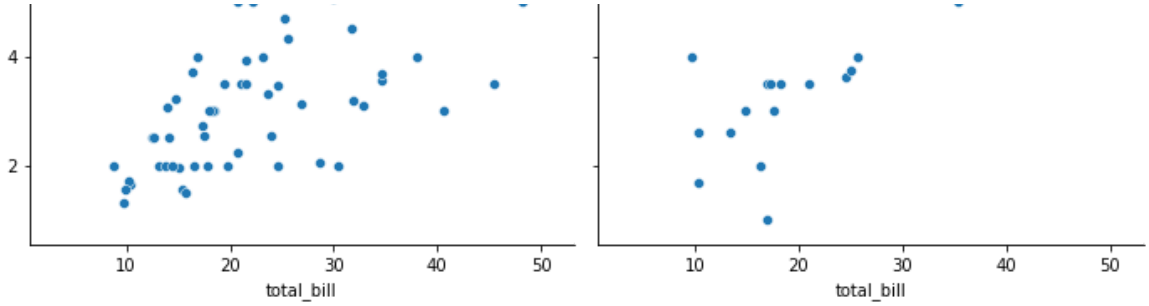
Out[26]:
```
<seaborn.axisgrid.FacetGrid at 0x1a0e2a9e0a0>
```



In [27]:
```
sns.relplot(data=tips,x="total_bill",y="tip",kind="scatter",col="sex",row="day")
```

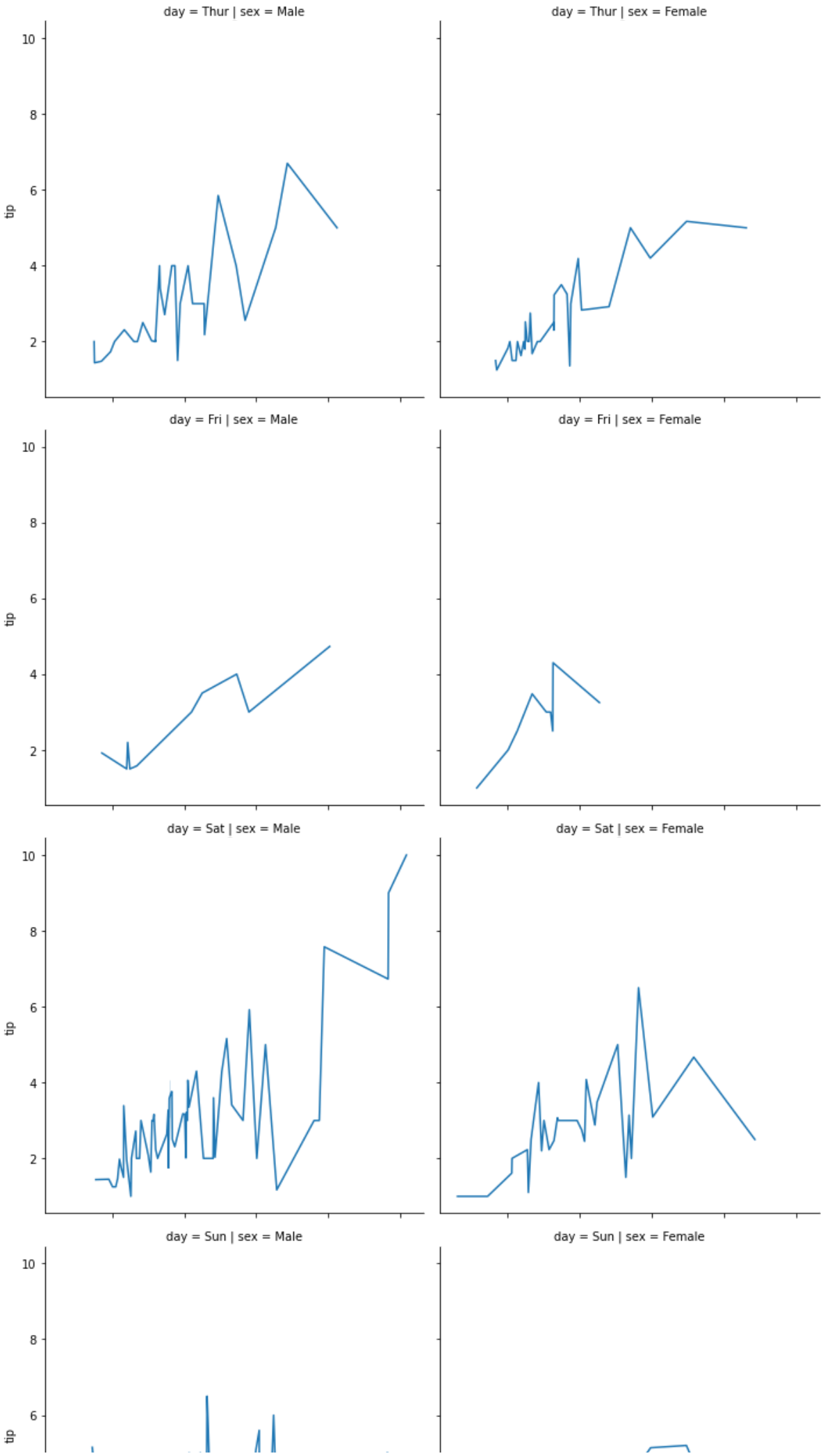
Out[27]:
```
<seaborn.axisgrid.FacetGrid at 0x1a0e2ac8dc0>
```

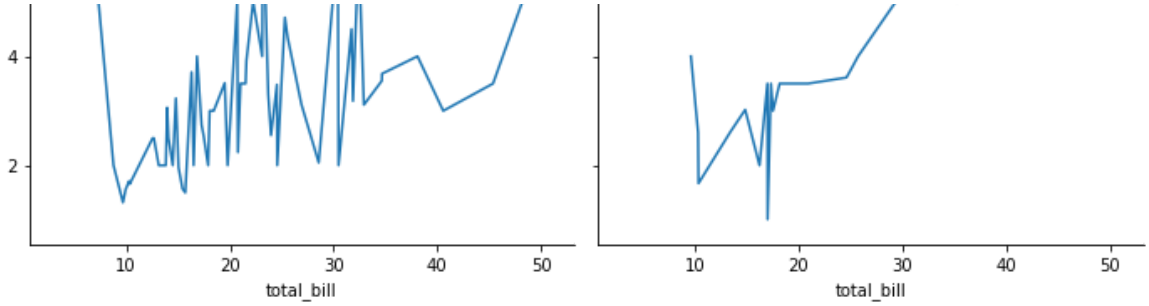

day = Thur | sex = Male
day = Thur | sex = Female
day = Fri | sex = Male
day = Fri | sex = Female
day = Sat | sex = Male
day = Sat | sex = Female
day = Sun | sex = Male
day = Sun | sex = Female
tip
10
8
6
4
2



In [28]:
```
sns.relplot(data=tips,x="total_bill",y="tip",kind="line",col="sex",row="day")
```

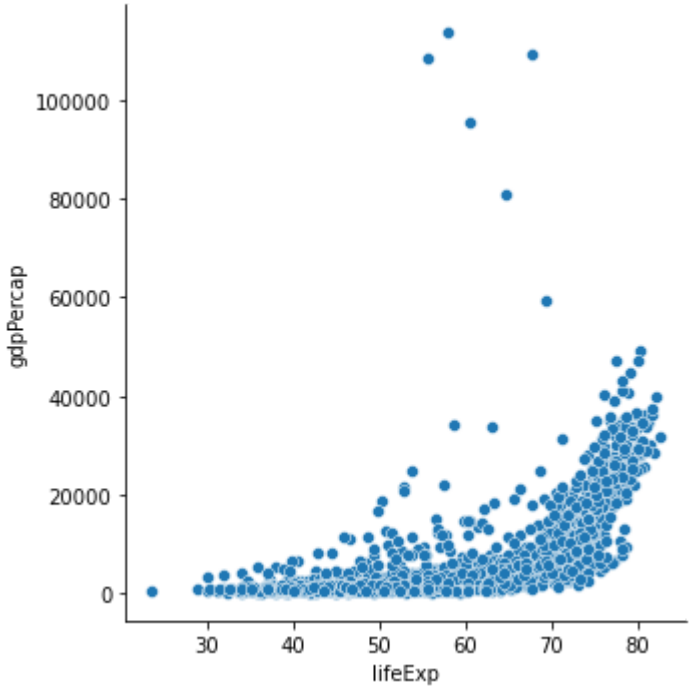
Out[28]:
```
<seaborn.axisgrid.FacetGrid at 0x1a0e2a9f190>
```


day = Thur | sex = Male
day = Thur | sex = Female
day = Fri | sex = Male
day = Fri | sex = Female
day = Sat | sex = Male
day = Sat | sex = Female
day = Sun | sex = Male
day = Sun | sex = Female
tip
10
8
6
4
2



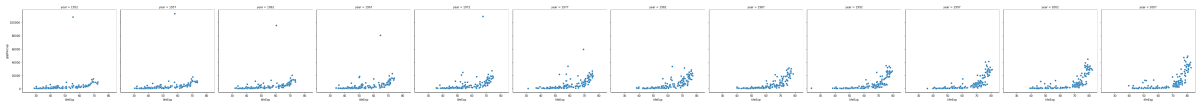
In [29]:
```
#col wrap
sns.relplot(data=gap,x="lifeExp",y="gdpPercap",kind="scatter")
```

Out[29]:
```
<seaborn.axisgrid.FacetGrid at 0x1a0e4195040>
```



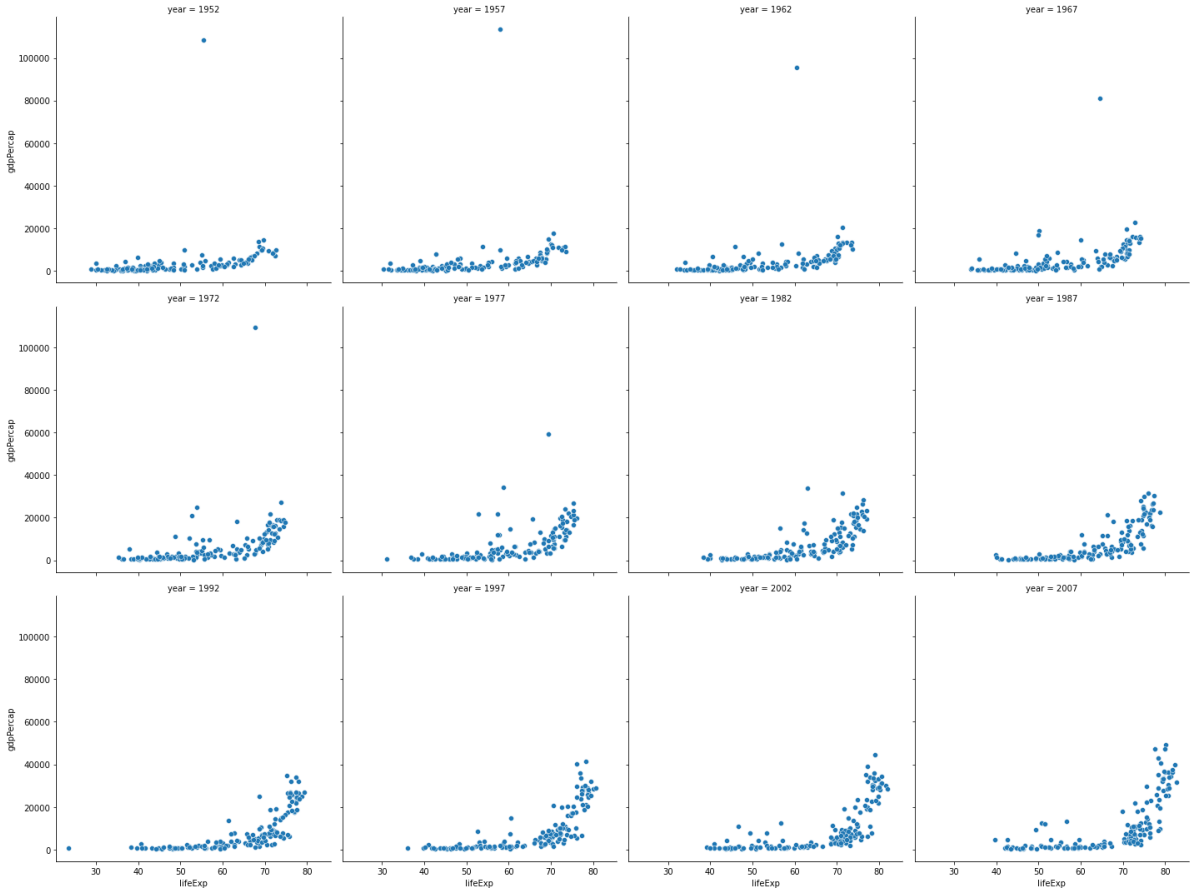
In [30]:
```
sns.relplot(data=gap,x="lifeExp",y="gdpPercap",kind="scatter",col="year")
```

Out[30]:
```
<seaborn.axisgrid.FacetGrid at 0x1a0e41ff820>
```



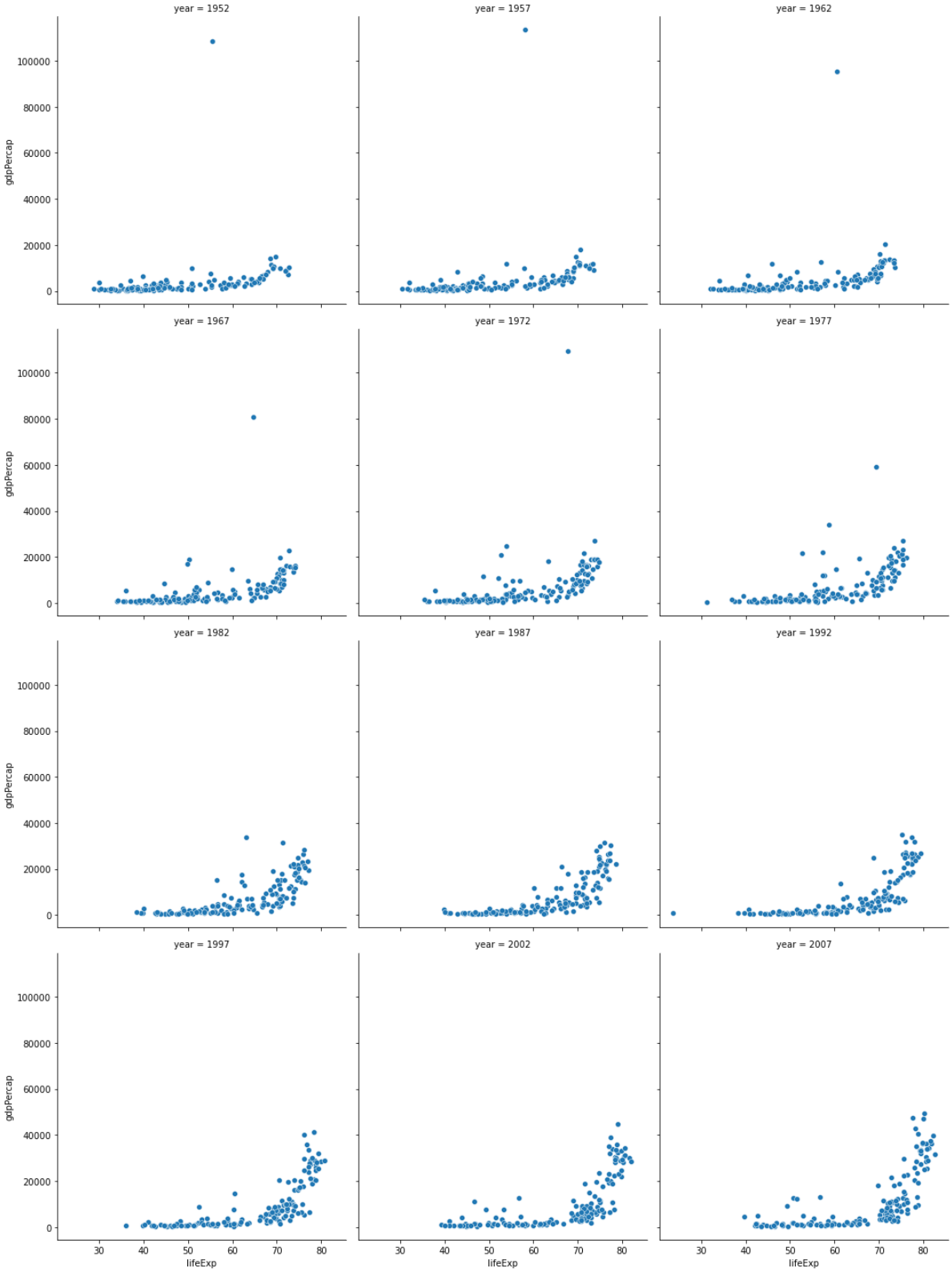
In [31]:
```
sns.relplot(data=gap,x="lifeExp",y="gdpPercap",kind="scatter",col="year",col_wrap=4
```

Out[31]:
```
<seaborn.axisgrid.FacetGrid at 0x1a0e4681f10>
```



In [32]:
```
sns.relplot(data=gap,x="lifeExp",y="gdpPercap",kind="scatter",col="year",col_wrap=
```

Out[32]:
```
<seaborn.axisgrid.FacetGrid at 0x1a0e70828e0>
```



2.Distribution Plot -used for univariate analysis -used to find out the distribution -Range of the observation -central tendency -is the data bimodal? -are there outliers?

Plots under distribution plot -histplot -kdeplot -rugplot

In [33]:
```
#figure level->displot
#axes level->histplot->kdeplot->rugplot
```

In [34]:
```
#plotting univariate histogram
tips
```

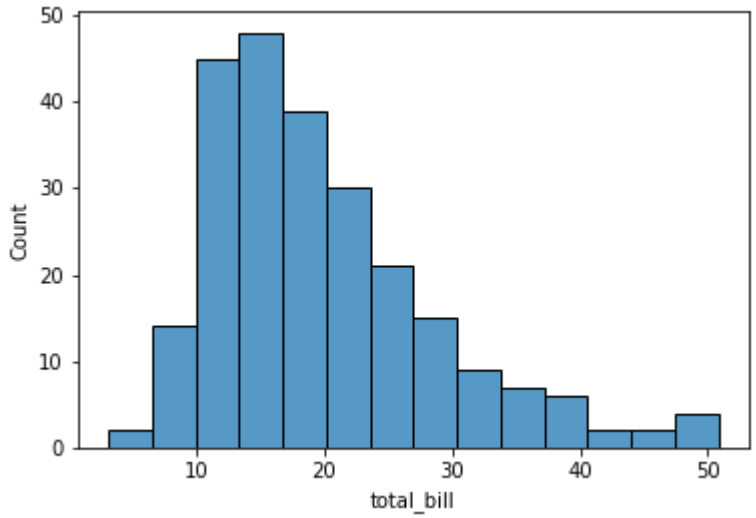
Out[34]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
...	...	...	...	...	...	...	...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	2.00	Male	Yes	Sat	Dinner	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

244 rows × 7 columns

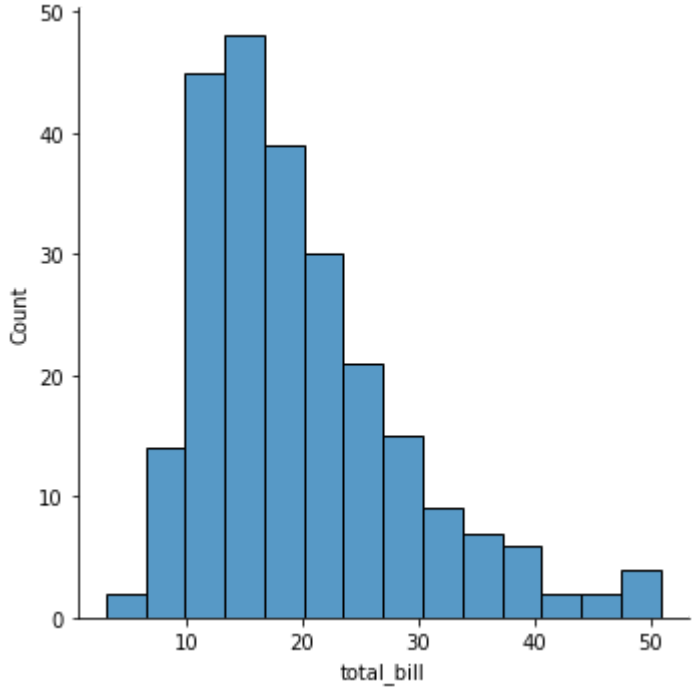
In [35]:
```
sns.histplot(data=tips,x="total_bill")
```

Out[35]:
```
<AxesSubplot:xlabel='total_bill', ylabel='Count'>
```



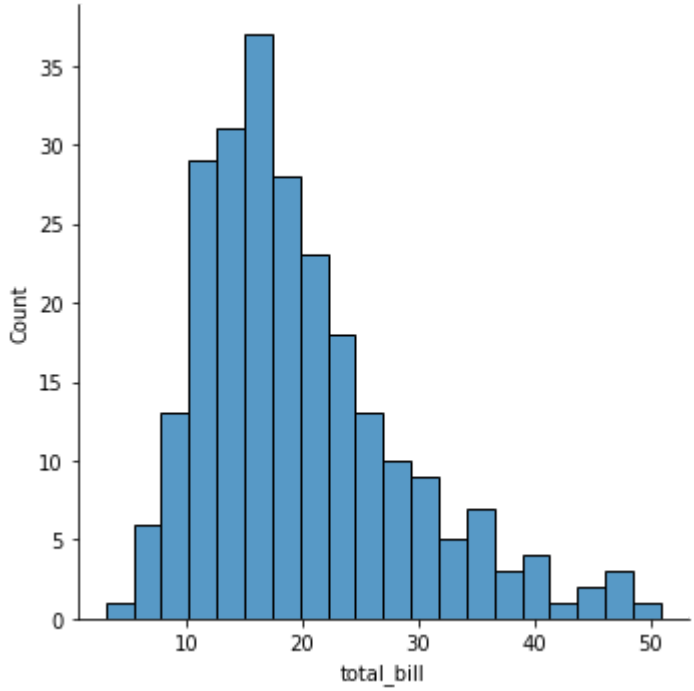
In [36]:
```
sns.displot(data=tips,x="total_bill",kind="hist")
```

Out[36]:
```
<seaborn.axisgrid.FacetGrid at 0x1a0e87be550>
```



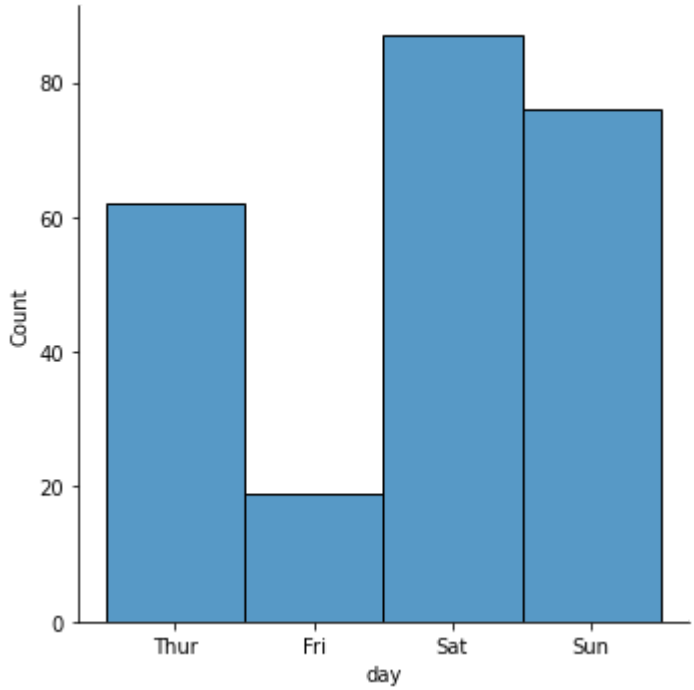
In [37]:
```
#bins parameter
sns.displot(data=tips,x="total_bill",kind="hist",bins=20)
```

Out[37]:
```
<seaborn.axisgrid.FacetGrid at 0x1a0e8160e50>
```



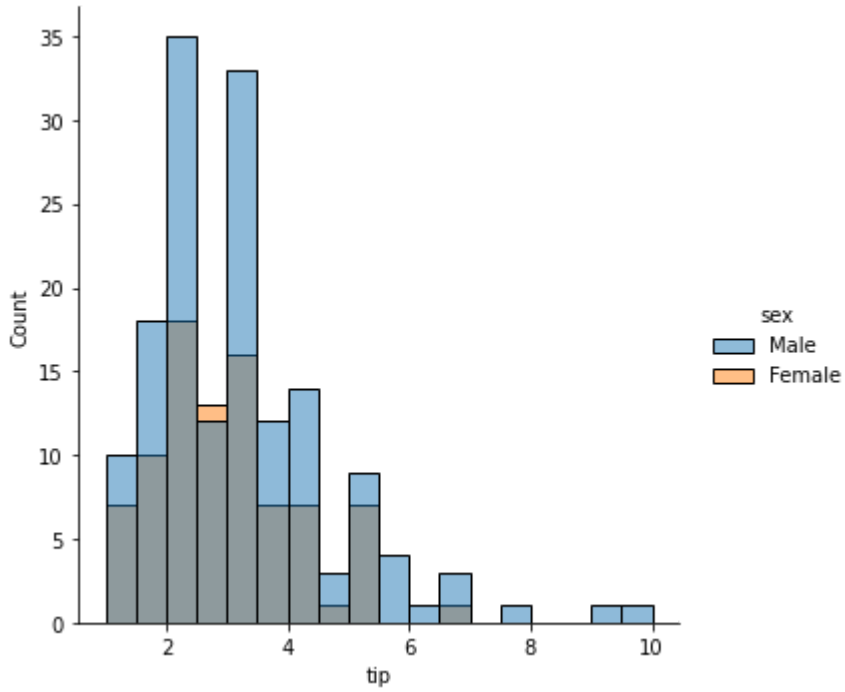
In [38]:
```
#its also possible to visualize the distribution of a categorical variable using th
#Discreate bins are automatically set for categorical variables.
sns.displot(data=tips,x="day",kind="hist")
#countplot
```

Out[38]:
```
<seaborn.axisgrid.FacetGrid at 0x1a0e81353d0>
```



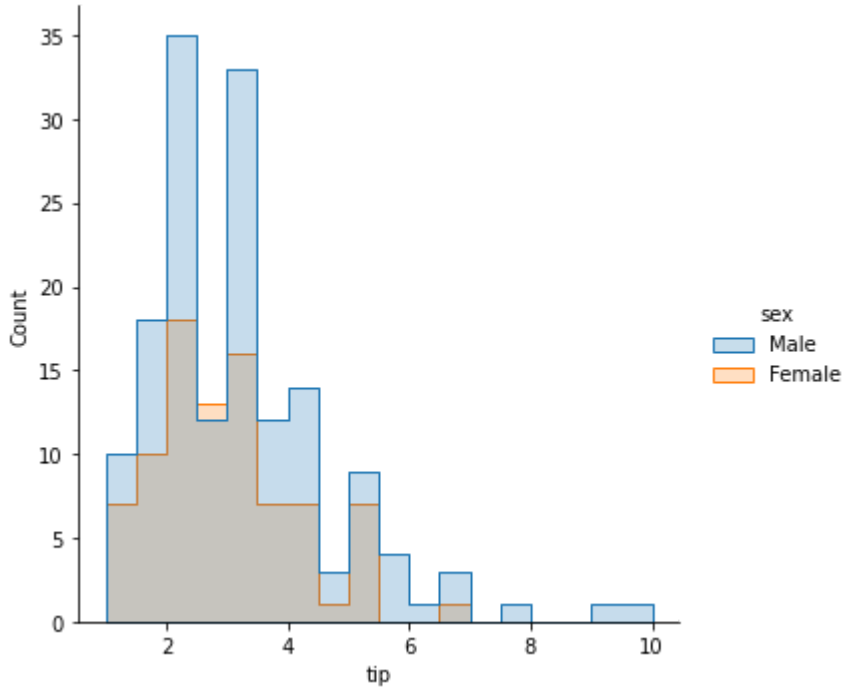
In [39]:
```
#hue parameter
sns.displot(data=tips,x="tip",kind="hist",hue="sex")
```

Out[39]:
```
<seaborn.axisgrid.FacetGrid at 0x1a0e882c7f0>
```



In [40]:
```
#element->step
sns.displot(data=tips,x="tip",kind="hist",hue="sex",element="step")
```

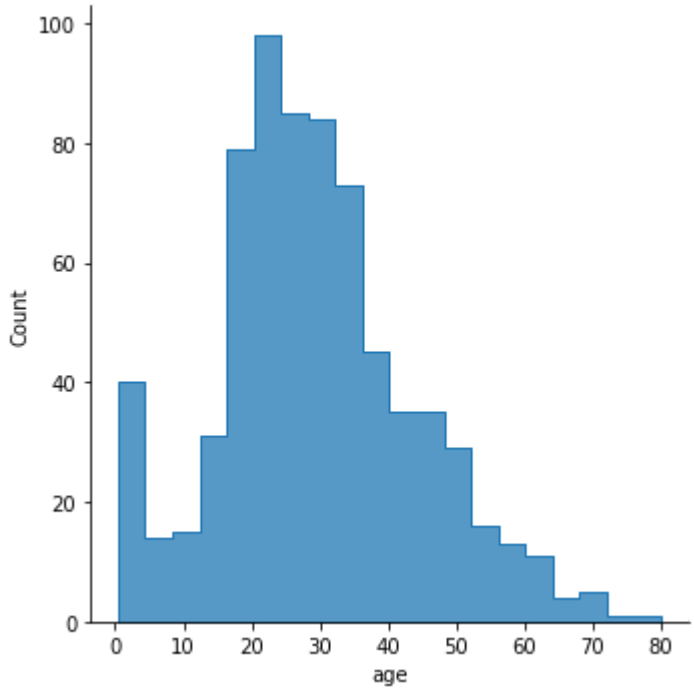
Out[40]:
```
<seaborn.axisgrid.FacetGrid at 0x1a0e8135310>
```



In [41]:
```
titanic=sns.load_dataset("titanic")
```

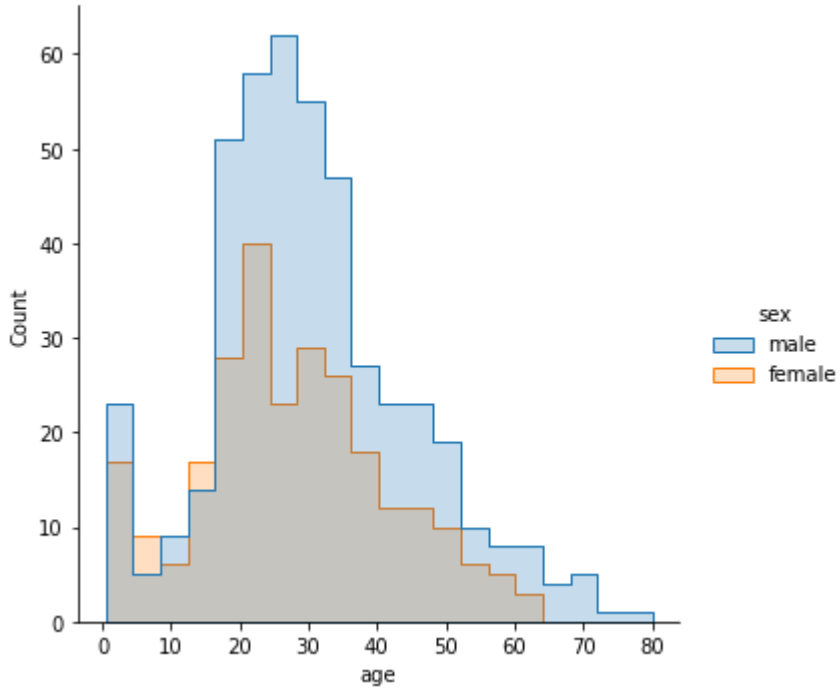
In [42]:
```
sns.displot(data=titanic,x="age",kind="hist",element="step")
```

Out[42]:
```
<seaborn.axisgrid.FacetGrid at 0x1a0e8a56b50>
```



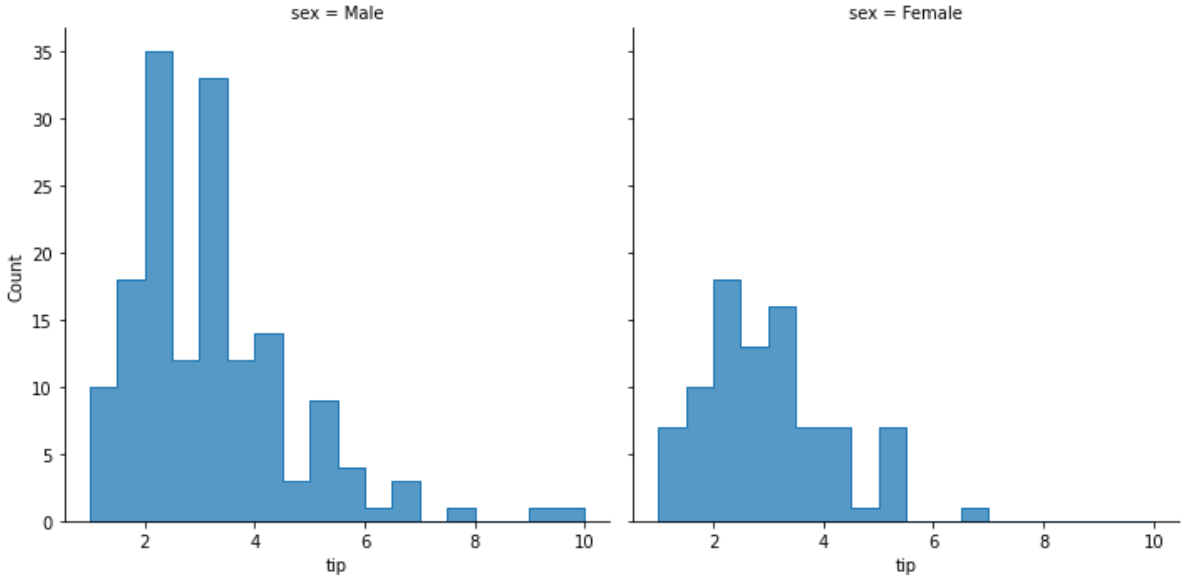
In [43]:
```
sns.displot(data=titanic,x="age",kind="hist",element="step",hue="sex")
```

Out[43]:
```
<seaborn.axisgrid.FacetGrid at 0x1a0e89981f0>
```



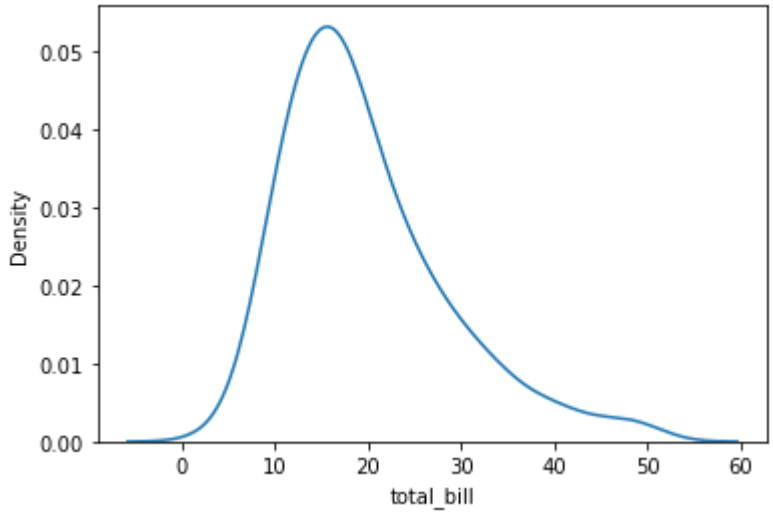
In [44]:
```
#faceting using col and row but will not work on histplot function.
sns.displot(data=tips,x="tip",kind="hist",element="step",col="sex")
```

Out[44]:
```
<seaborn.axisgrid.FacetGrid at 0x1a0e8914bb0>
```



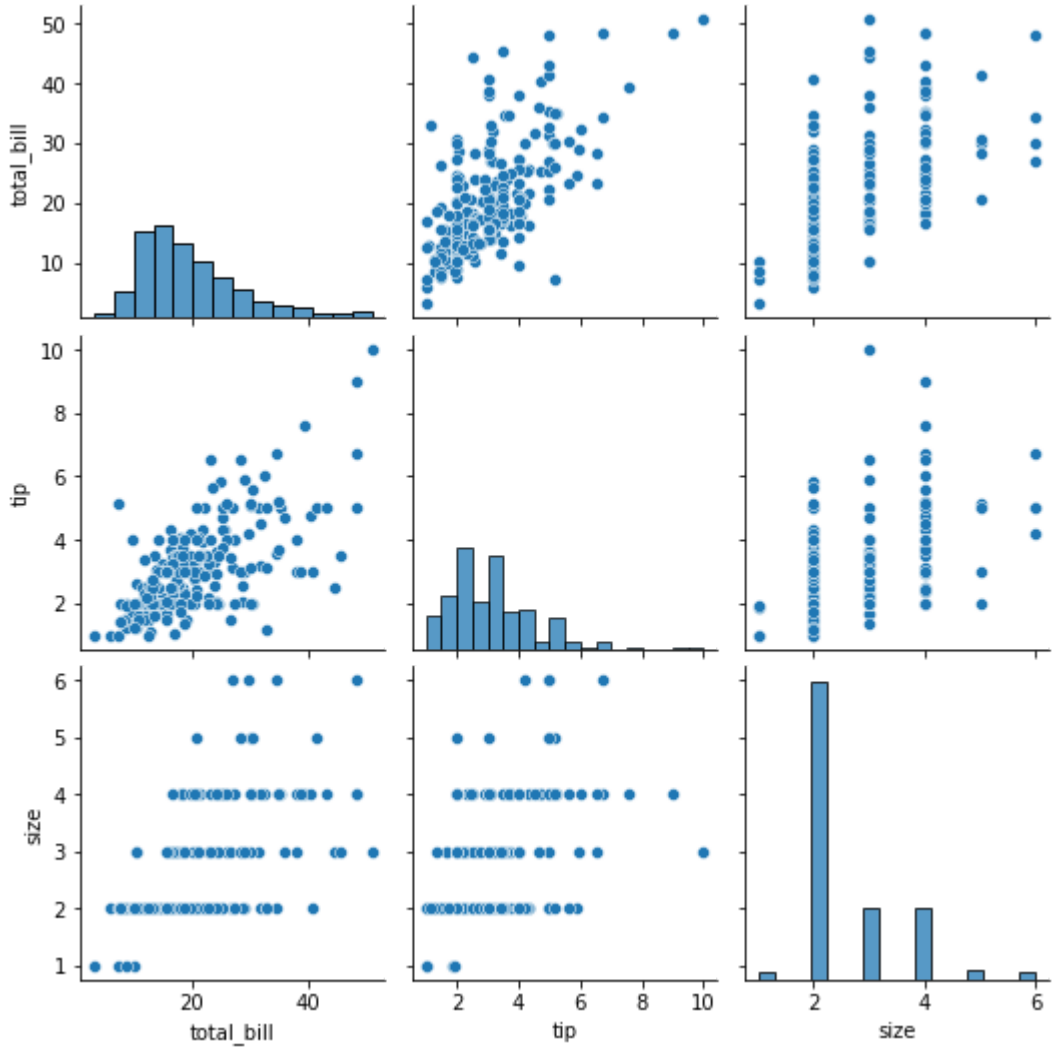
In [45]:
```
#kdeplot
#rather than using discrete bins,a KDE plot smooths the observation with a gaussiar
sns.kdeplot(data=tips,x="total_bill")
```

Out[45]:
```
<AxesSubplot:xlabel='total_bill', ylabel='Density'>
```



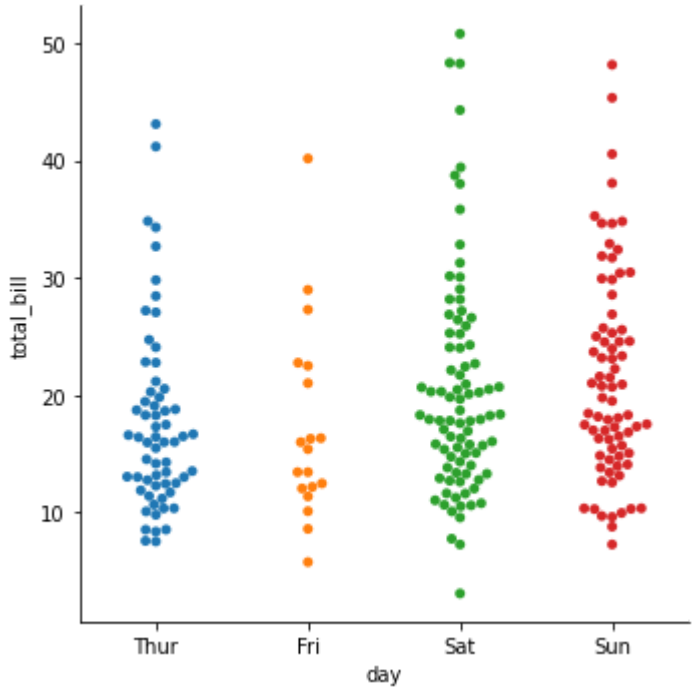
In [46]:
```
sns.pairplot(tips)
```

Out[46]:
```
<seaborn.axisgrid.PairGrid at 0x1a0e9bc64f0>
```



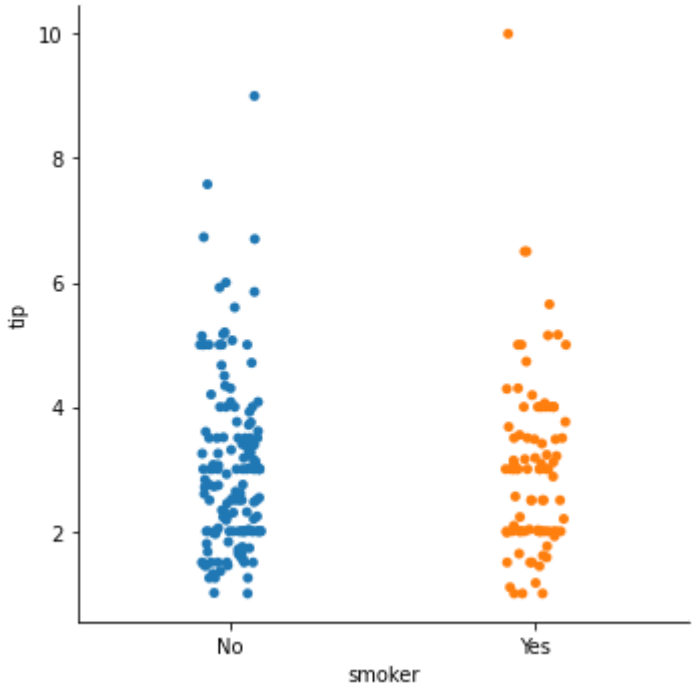
In [47]:
```
sns.catplot(data=tips,x="day",y="total_bill",kind="swarm")
```

Out[47]:
```
<seaborn.axisgrid.FacetGrid at 0x1a0ea596f10>
```



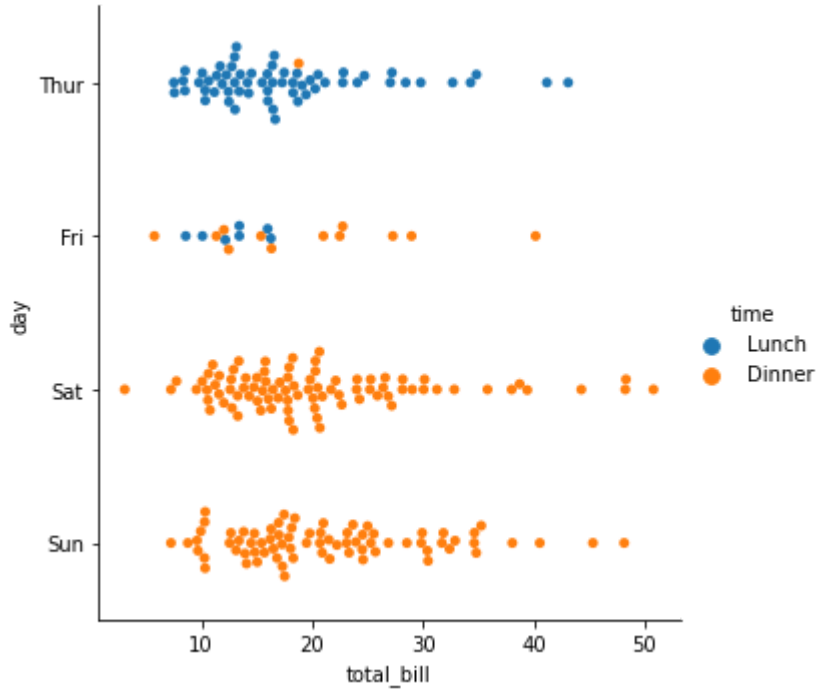
In [48]:
```
sns.catplot(data=tips, x="smoker", y="tip", order=["No", "Yes"])
```

Out[48]:
```
<seaborn.axisgrid.FacetGrid at 0x1a0e9b99910>
```



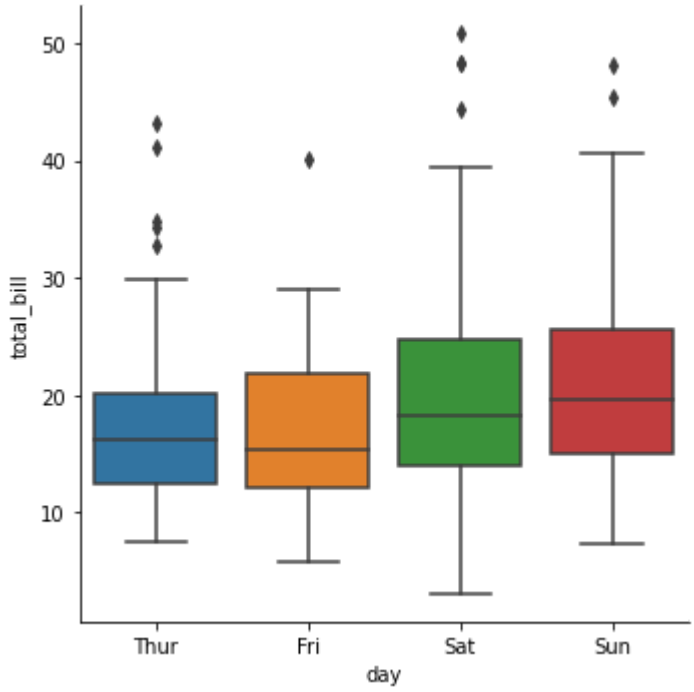
In [49]:
```
sns.catplot(data=tips, x="total_bill", y="day", hue="time", kind="swarm")
```

Out[49]:
```
<seaborn.axisgrid.FacetGrid at 0x1a0ea6008e0>
```



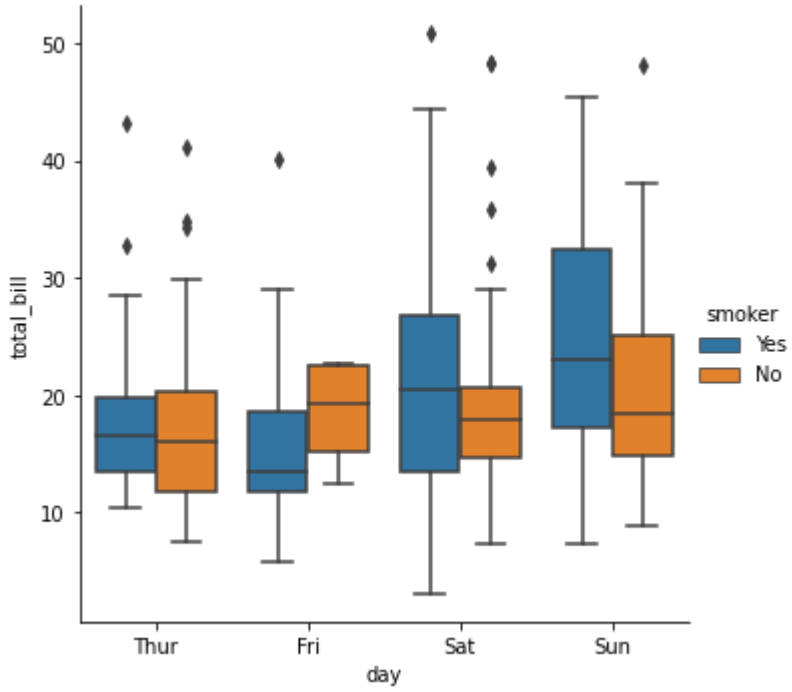
In [50]:
```
sns.catplot(data=tips, x="day", y="total_bill", kind="box")
```

Out[50]:
```
<seaborn.axisgrid.FacetGrid at 0x1a0ea868910>
```



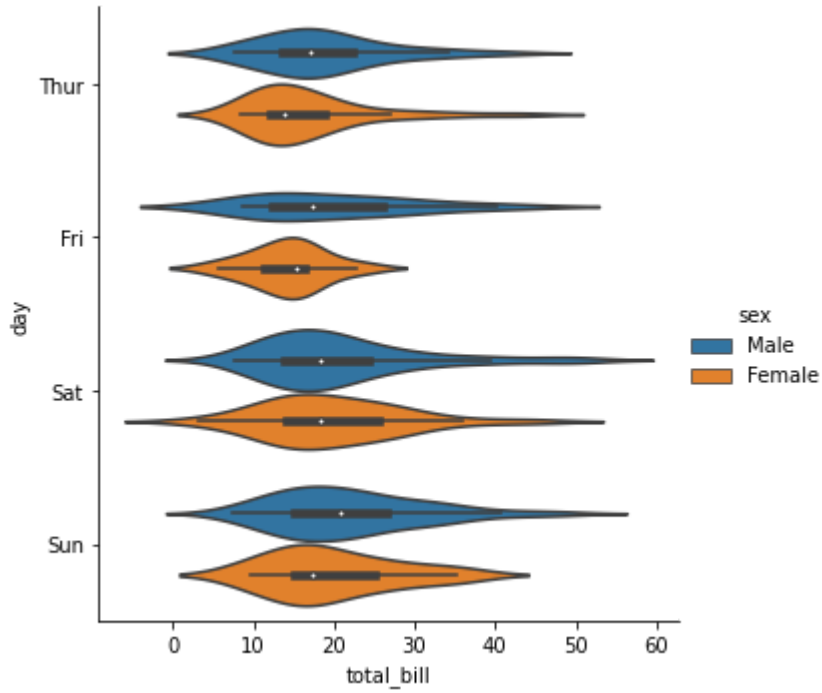
In [51]:
```
sns.catplot(data=tips, x="day", y="total_bill", hue="smoker", kind="box")
```

Out[51]:
```
<seaborn.axisgrid.FacetGrid at 0x1a0ea527d60>
```

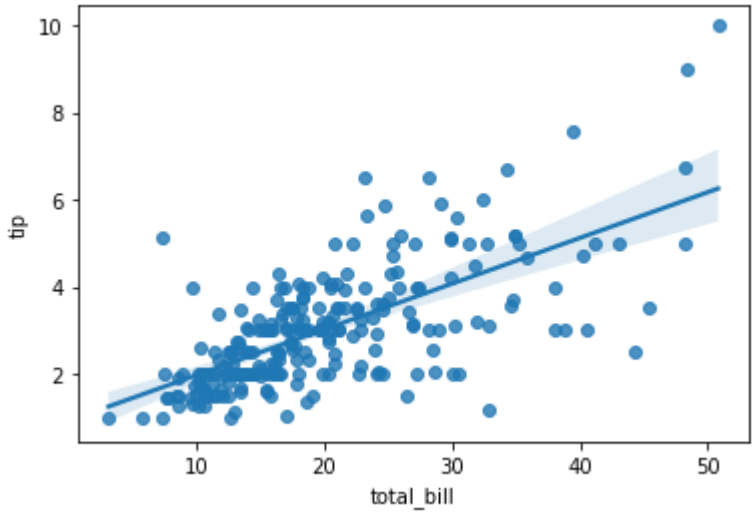


In [52]:
```
sns.catplot(
    data=tips, x="total_bill", y="day", hue="sex", kind="violin",)
```

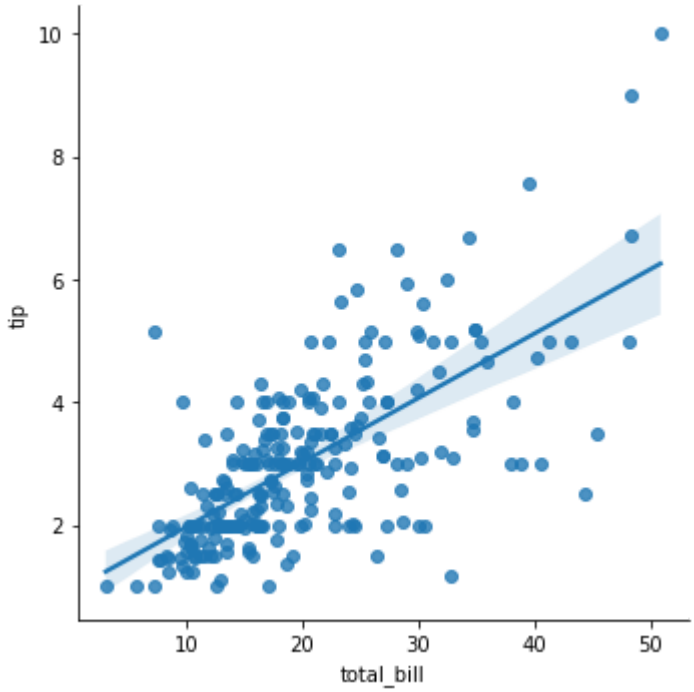
Out[52]:
```
<seaborn.axisgrid.FacetGrid at 0x1a0ea8ca190>
```



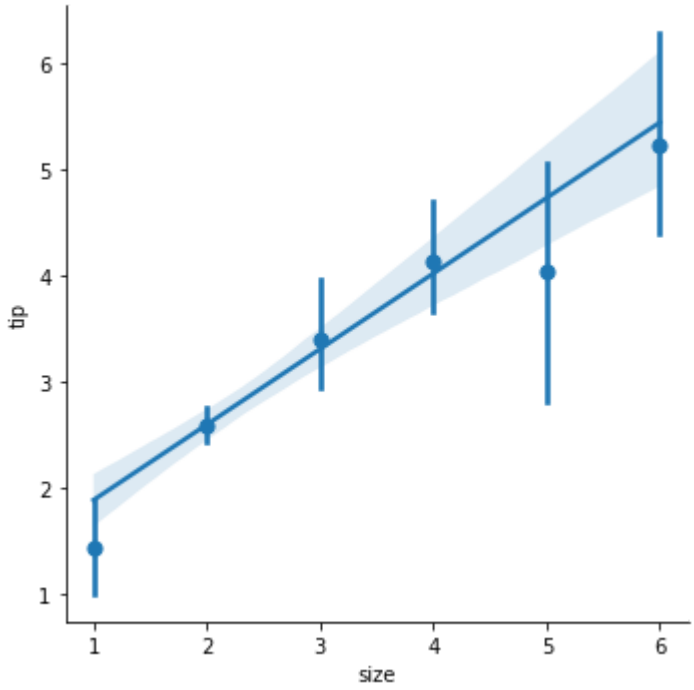
In [53]:
```
tips = sns.load_dataset("tips")
sns.regplot(x="total_bill", y="tip", data=tips);
```



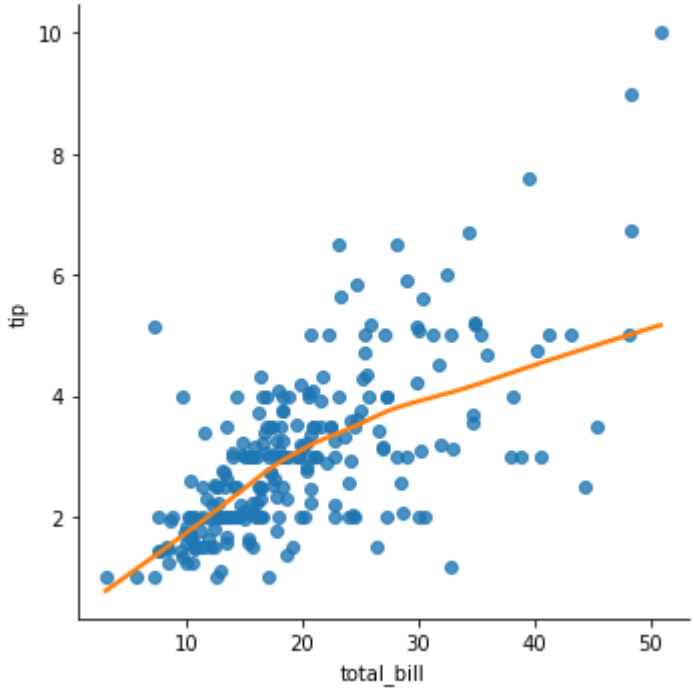
In [54]:
```
sns.lmplot(x="total_bill", y="tip", data=tips);
```



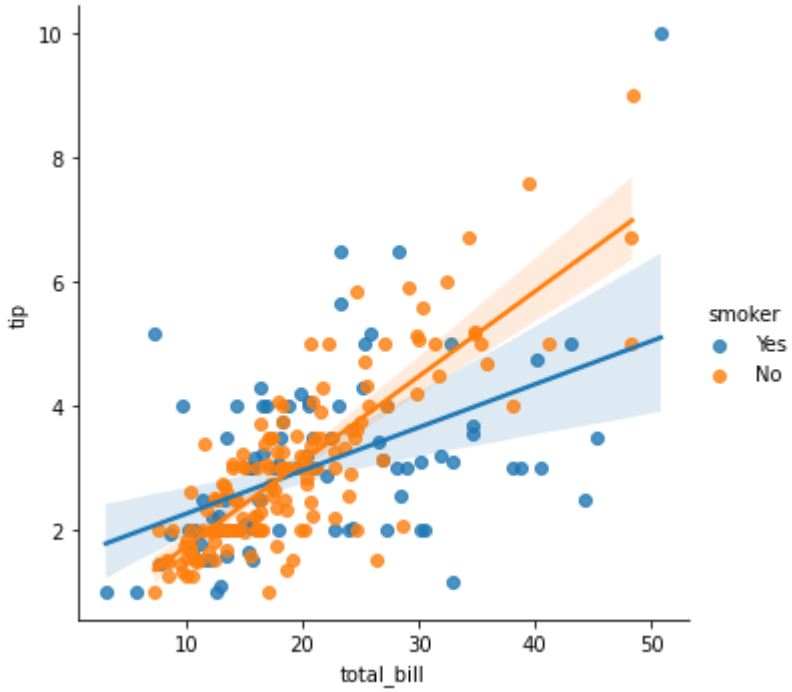
In [55]:
```
import numpy as np
sns.lmplot(x="size", y="tip", data=tips, x_estimator=np.mean);
```



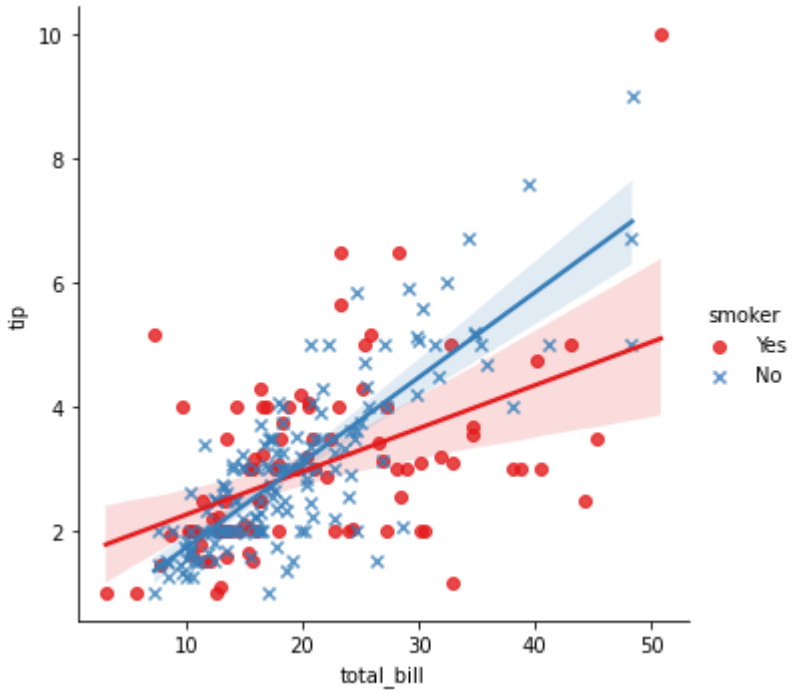
In [56]:
```
sns.lmplot(x="total_bill", y="tip", data=tips,
           lowess=True, line_kws={"color": "C1"});
```



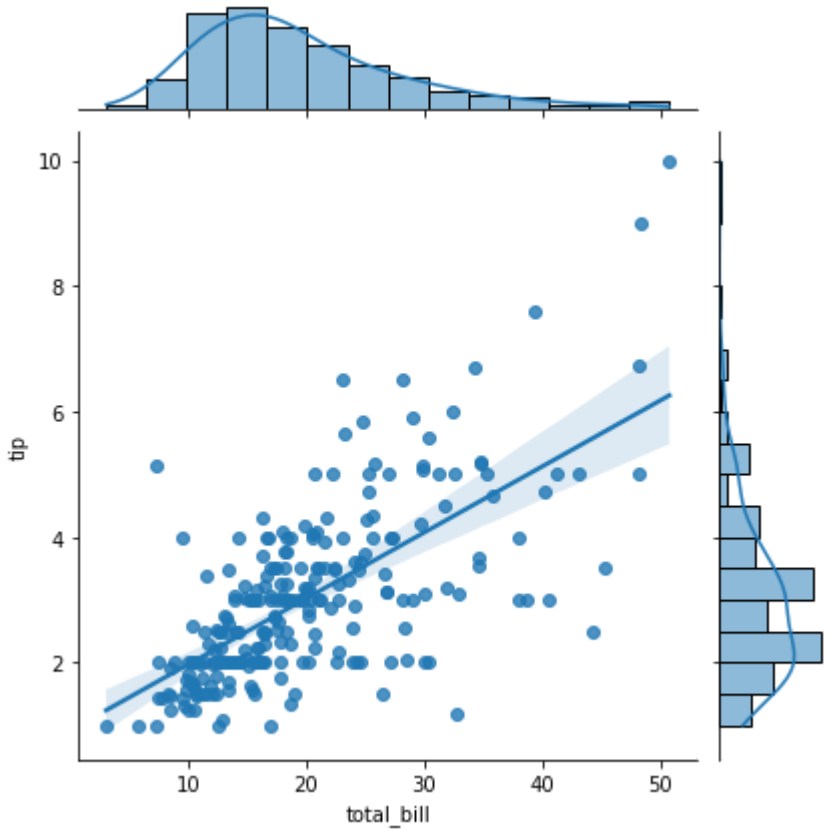
In [57]:
```
sns.lmplot(x="total_bill", y="tip", hue="smoker", data=tips);
```



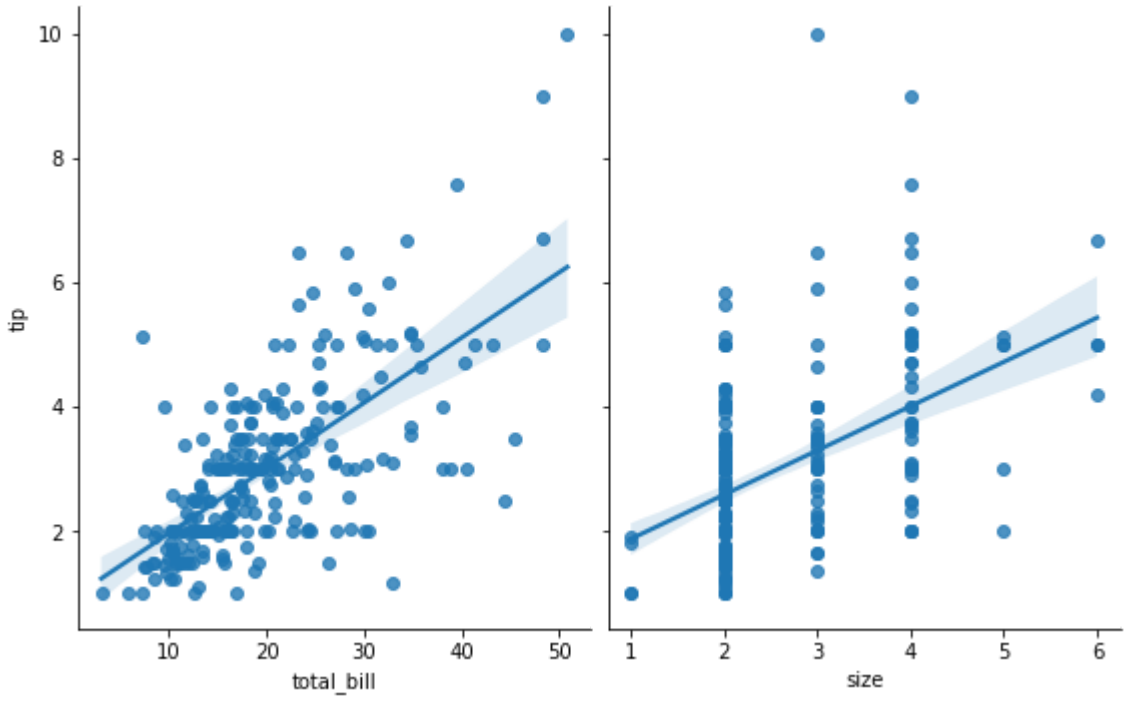
In [58]:
```
sns.lmplot(x="total_bill", y="tip", hue="smoker", data=tips,
           markers=["o", "x"], palette="Set1");
```



In [59]:
```
sns.jointplot(x="total_bill", y="tip", data=tips, kind="reg");
```

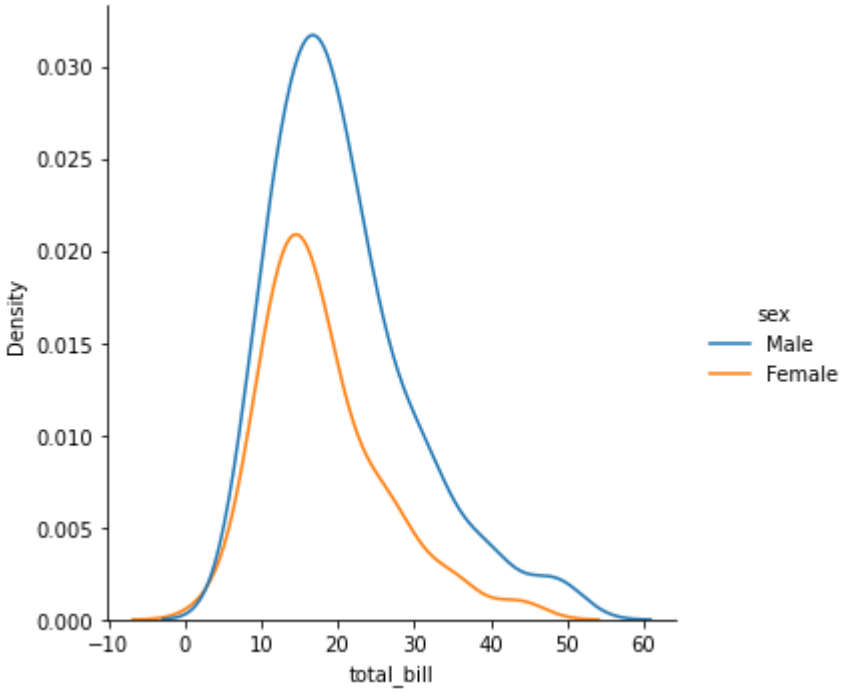


In [60]:
```
sns.pairplot(tips, x_vars=["total_bill", "size"], y_vars=["tip"],
             height=5, aspect=.8, kind="reg");
```



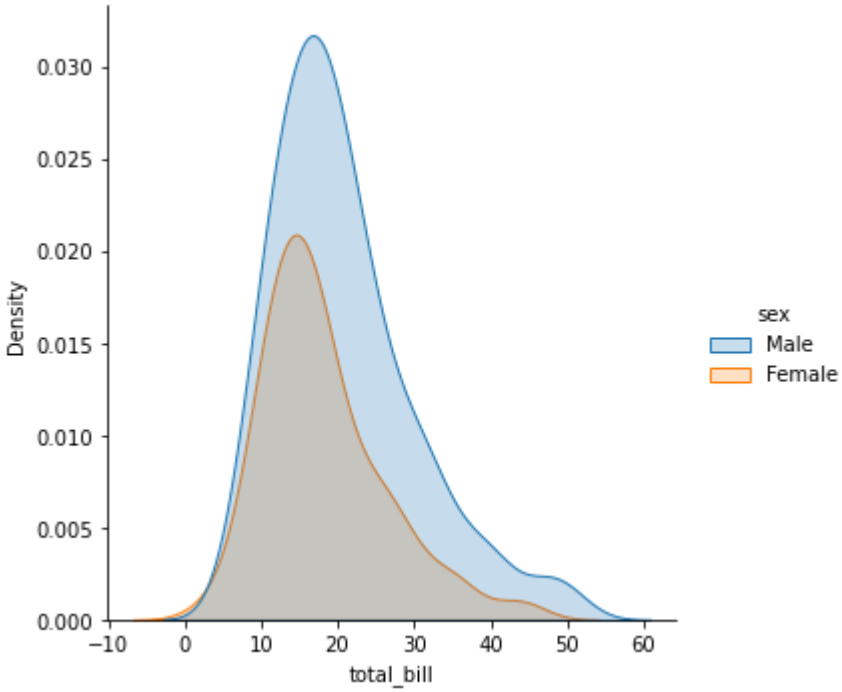
In [61]:
```
#hue->fill
sns.displot(data=tips,x="total_bill",kind="kde",hue="sex")
```

Out[61]:
```
<seaborn.axisgrid.FacetGrid at 0x1a0ebd575e0>
```



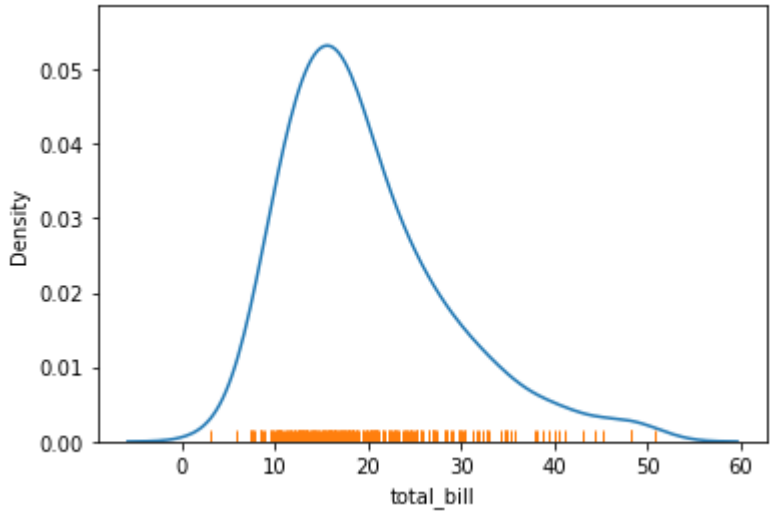
In [62]:
```
sns.displot(data=tips,x="total_bill",kind="kde",hue="sex",fill=True)
```

Out[62]:
```
<seaborn.axisgrid.FacetGrid at 0x1a0ebfbad00>
```



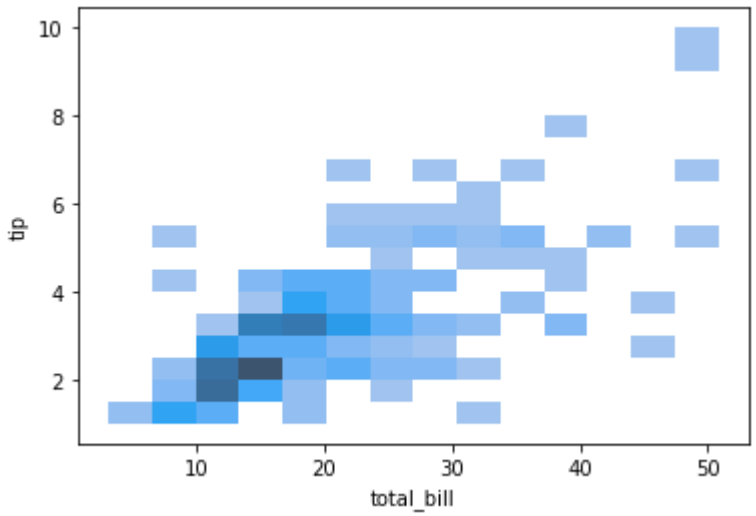
In [63]:
```
#Rugplot
#Plot marginal distribution by drawing ticks along the x and y axes.
#This function is intended to complement other plots by showing the location of ind
sns.kdeplot(data=tips,x="total_bill")
sns.rugplot(data=tips,x="total_bill")
```

Out[63]:
```
<AxesSubplot:xlabel='total_bill', ylabel='Density'>
```



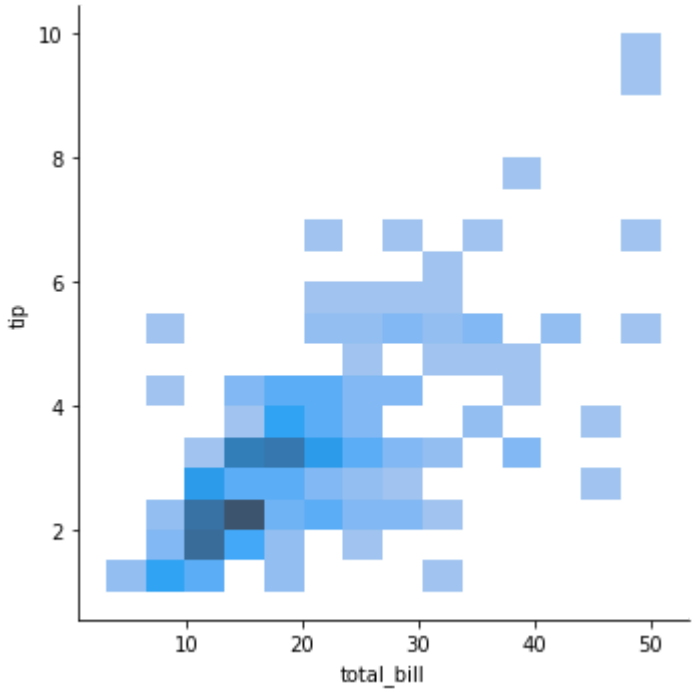
In [64]:
```
#Bivariate Histogram
#A bivariate histogram bins the data within rectangles that tile the plot
#and then shows the count of observation within each rectangle with the fill colour

sns.histplot(data=tips,x="total_bill",y="tip")
```

Out[64]:
```
<AxesSubplot:xlabel='total_bill', ylabel='tip'>
```



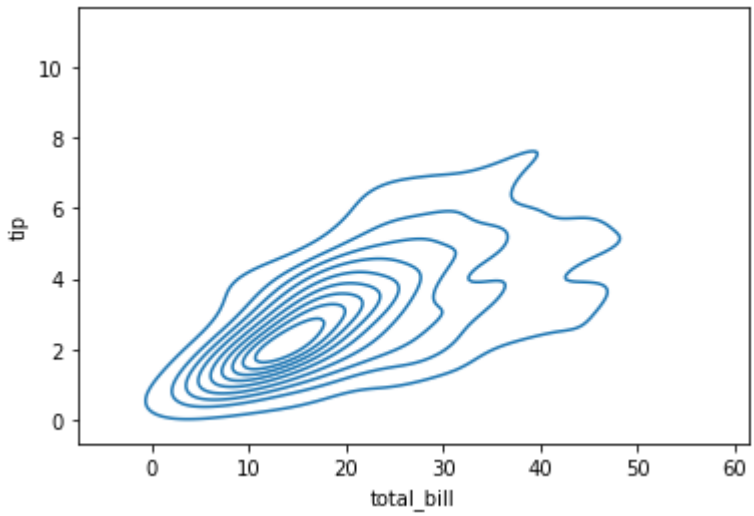
In [65]:
```
sns.displot(data=tips,x="total_bill",y="tip",kind="hist")
```

Out[65]:
```
<seaborn.axisgrid.FacetGrid at 0x1a0ec05ef70>
```



In [66]:
```
#Bivariate kdeplot
#a bivariate kde plot smoothe the (x,y) observation with a 2D gaussian.
sns.kdeplot(data=tips,x="total_bill",y="tip")
```

Out[66]:
```
<AxesSubplot:xlabel='total_bill', ylabel='tip'>
```



3.Matrix Plot -Heatmap -Clustermap

In [67]:
```
#heatmap
#plot rectangular data as a colour encoded matrix.
gap
```

Out[67]:

	country	continent	year	lifeExp	pop	gdpPercap	iso_alpha	iso_num
0	Afghanistan	Asia	1952	28.801	8425333	779.445314	AFG	4
1	Afghanistan	Asia	1957	30.332	9240934	820.853030	AFG	4
2	Afghanistan	Asia	1962	31.997	10267083	853.100710	AFG	4
3	Afghanistan	Asia	1967	34.020	11537966	836.197138	AFG	4
4	Afghanistan	Asia	1972	36.088	13079460	739.981106	AFG	4
...	...	...	...	...	...	...	...	...
1699	Zimbabwe	Africa	1987	62.351	9216418	706.157306	ZWE	716
1700	Zimbabwe	Africa	1992	60.377	10704340	693.420786	ZWE	716
1701	Zimbabwe	Africa	1997	46.809	11404948	792.449960	ZWE	716
1702	Zimbabwe	Africa	2002	39.989	11926563	672.038623	ZWE	716
1703	Zimbabwe	Africa	2007	43.487	12311143	469.709298	ZWE	716

1704 rows × 8 columns

In [68]:
```
temp_df=gap.pivot(index="country",columns="year",values="lifeExp")
```

In [69]:
```
temp_df
```

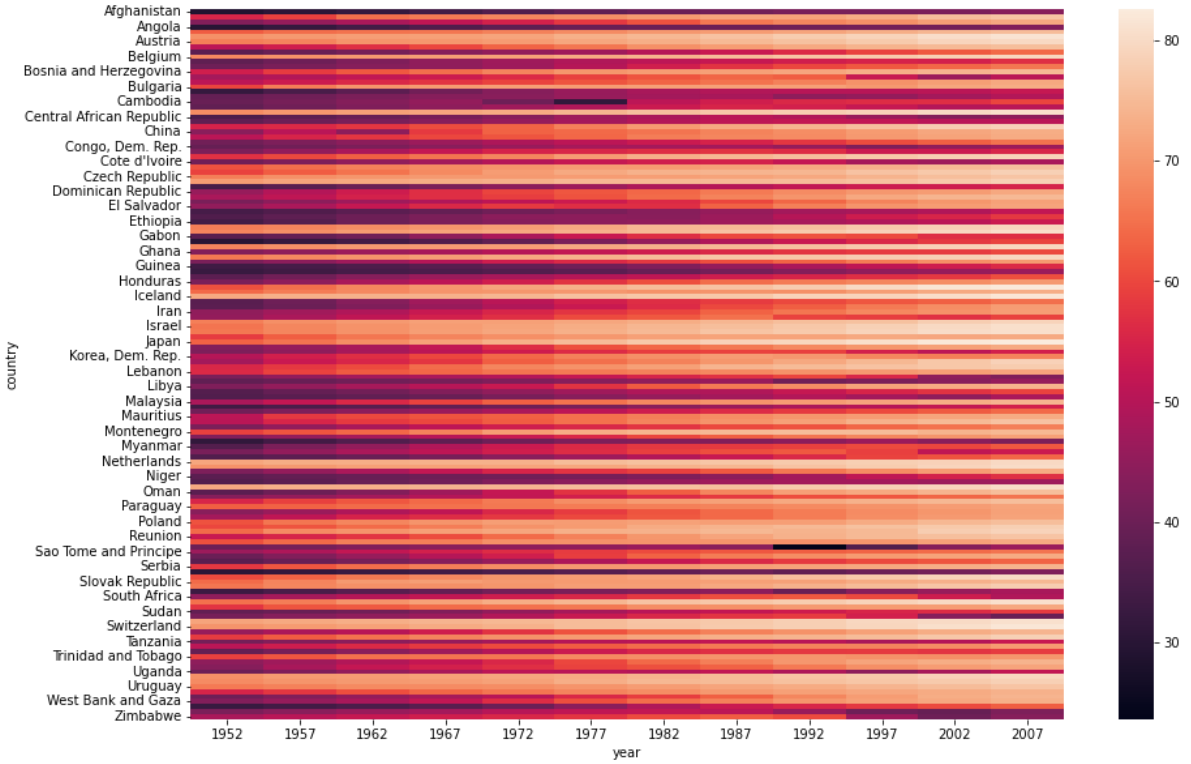
Out[69]:

year	1952	1957	1962	1967	1972	1977	1982	1987	1992	1997	2002
country											
Afghanistan	28.801	30.332	31.997	34.020	36.088	38.438	39.854	40.822	41.674	41.763	42.129
Albania	55.230	59.280	64.820	66.220	67.690	68.930	70.420	72.000	71.581	72.950	75.651
Algeria	43.077	45.685	48.303	51.407	54.518	58.014	61.368	65.799	67.744	69.152	70.994
Angola	30.015	31.999	34.000	35.985	37.928	39.483	39.942	39.906	40.647	40.963	41.003
Argentina	62.485	64.399	65.142	65.634	67.065	68.481	69.942	70.774	71.868	73.275	74.340
...	...	...	...	...	...	...	...	...	...	...	...
Vietnam	40.412	42.887	45.363	47.838	50.254	55.764	58.816	62.820	67.662	70.672	73.017
West Bank and Gaza	43.160	45.671	48.127	51.631	56.532	60.765	64.406	67.046	69.718	71.096	72.370
Yemen, Rep.	32.548	33.970	35.180	36.984	39.848	44.175	49.113	52.922	55.599	58.020	60.308
Zambia	42.038	44.077	46.023	47.768	50.107	51.386	51.821	50.821	46.100	40.238	39.193
Zimbabwe	48.451	50.469	52.358	53.995	55.635	57.674	60.363	62.351	60.377	46.809	39.989

142 rows × 12 columns

In [70]:
```
#axes level function
plt.figure(figsize=(15,10))
sns.heatmap(temp_df)
```

Out[70]:
```
<AxesSubplot:xlabel='year', ylabel='country'>
```



In [71]:
```
#annot
gap[gap["continent"]=="Europe"]
```

Out[71]:

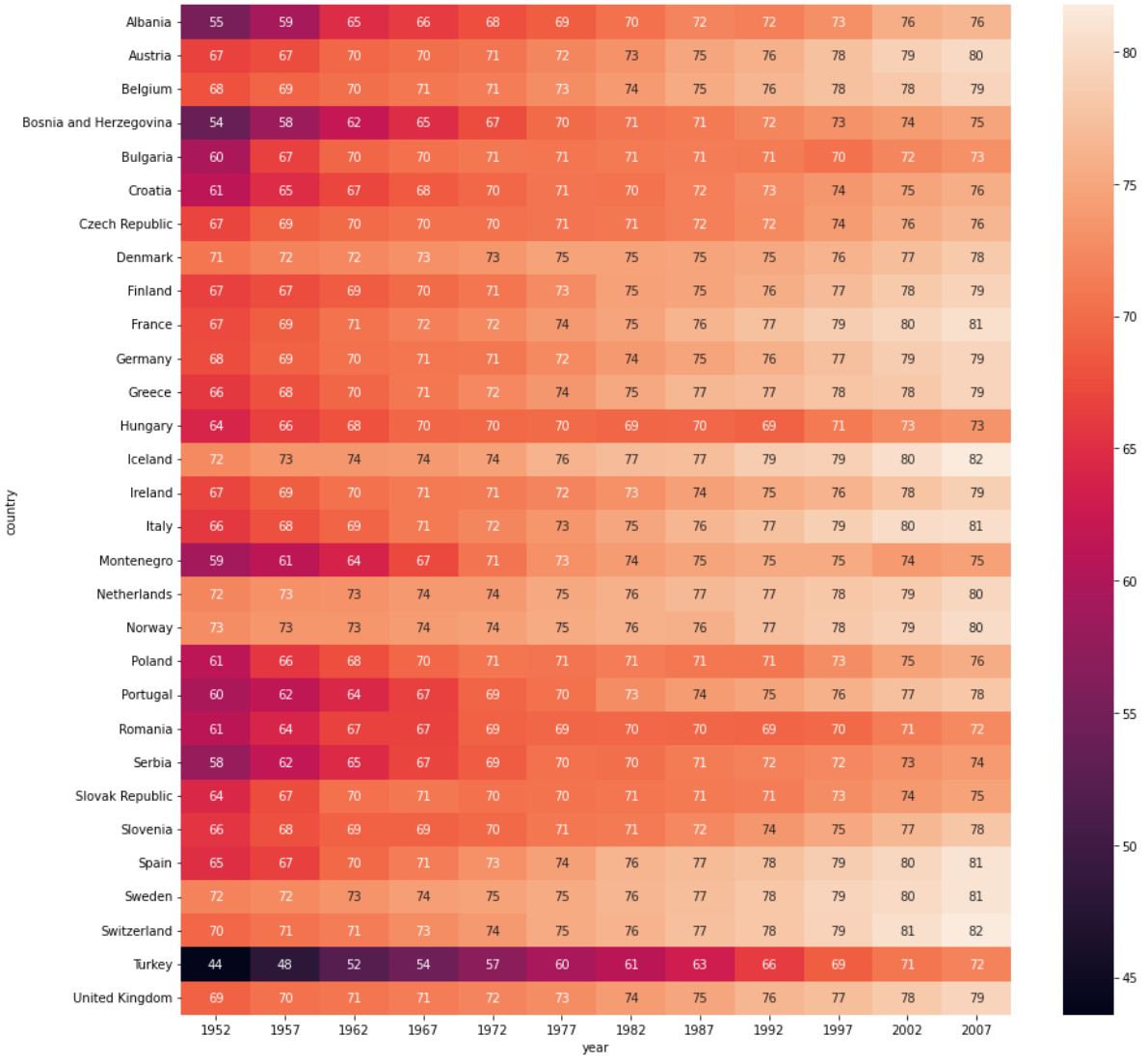
	country	continent	year	lifeExp	pop	gdpPercap	iso_alpha	iso_num
12	Albania	Europe	1952	55.230	1282697	1601.056136	ALB	8
13	Albania	Europe	1957	59.280	1476505	1942.284244	ALB	8
14	Albania	Europe	1962	64.820	1728137	2312.888958	ALB	8
15	Albania	Europe	1967	66.220	1984060	2760.196931	ALB	8
16	Albania	Europe	1972	67.690	2263554	3313.422188	ALB	8
...	...	...	...	...	...	...	...	...
1603	United Kingdom	Europe	1987	75.007	56981620	21664.787670	GBR	826
1604	United Kingdom	Europe	1992	76.420	57866349	22705.092540	GBR	826
1605	United Kingdom	Europe	1997	77.218	58808266	26074.531360	GBR	826
1606	United Kingdom	Europe	2002	78.471	59912431	29478.999190	GBR	826
1607	United Kingdom	Europe	2007	79.425	60776238	33203.261280	GBR	826

360 rows × 8 columns

In [72]:
```
temp_df=gap[gap["continent"]=="Europe"].pivot(index="country",columns="year",values
```

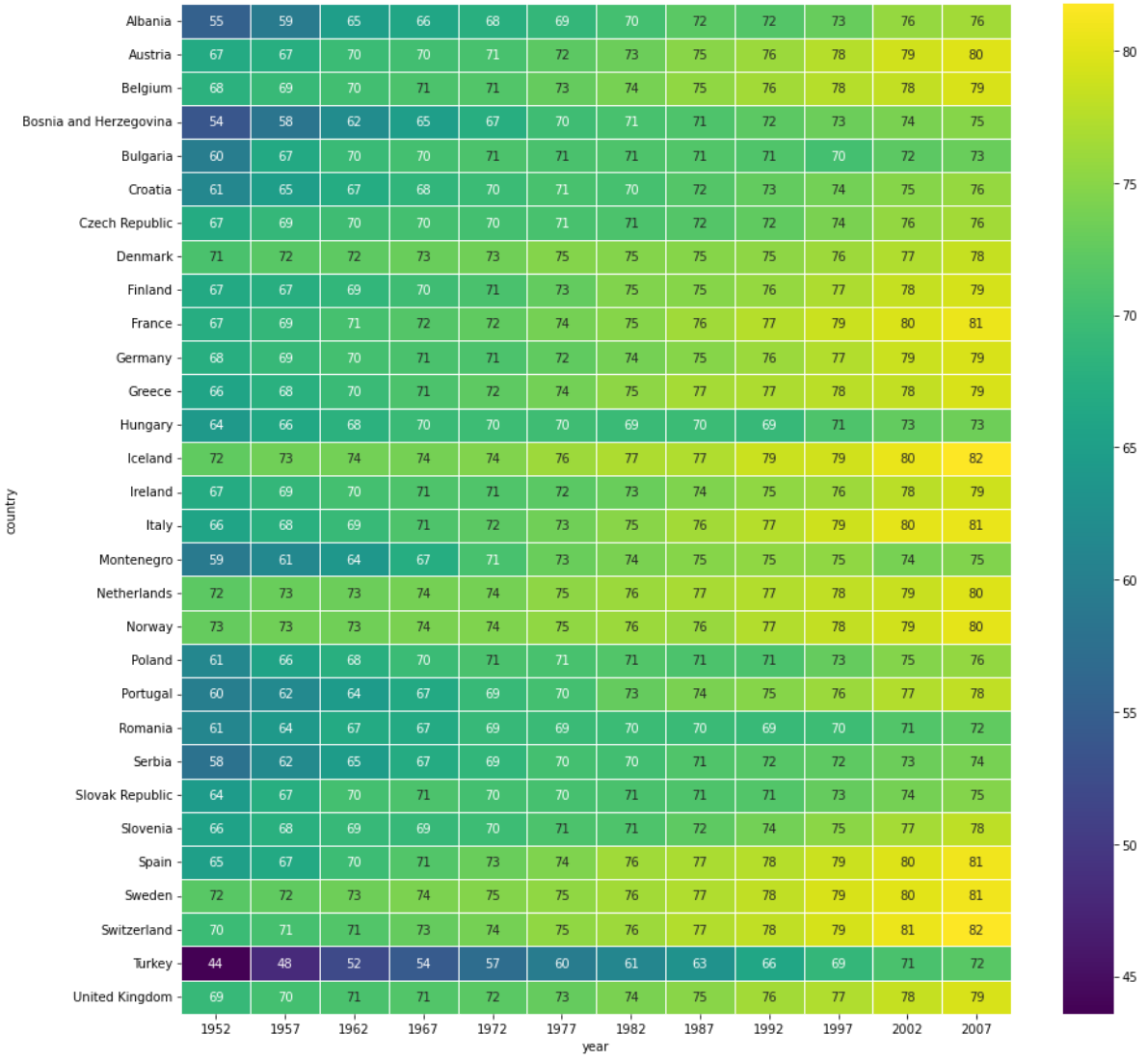
In [73]:
```
plt.figure(figsize=(15,15))
sns.heatmap(temp_df,annot=True)
```

Out[73]:
```
<AxesSubplot:xlabel='year', ylabel='country'>
```



In [74]:
```
plt.figure(figsize=(15,15))
sns.heatmap(temp_df,annot=True,linewidth=0.5,cmap="viridis")
```

Out[74]:
```
<AxesSubplot:xlabel='year', ylabel='country'>
```



In [75]:
```
plt.figure(figsize=(15,15))
sns.heatmap(temp_df,annot=True,linewidth=0.5,cmap="summer")
```

Out[75]:
```
<AxesSubplot:xlabel='year', ylabel='country'>
```

In [76]:
```
#Clustermap
#plot a matrix dataset as a hierarchically-clustered heatmap.
#this funtion requires scipy to be available.

iris=px.data.iris()
```

In [77]:
```
iris
```

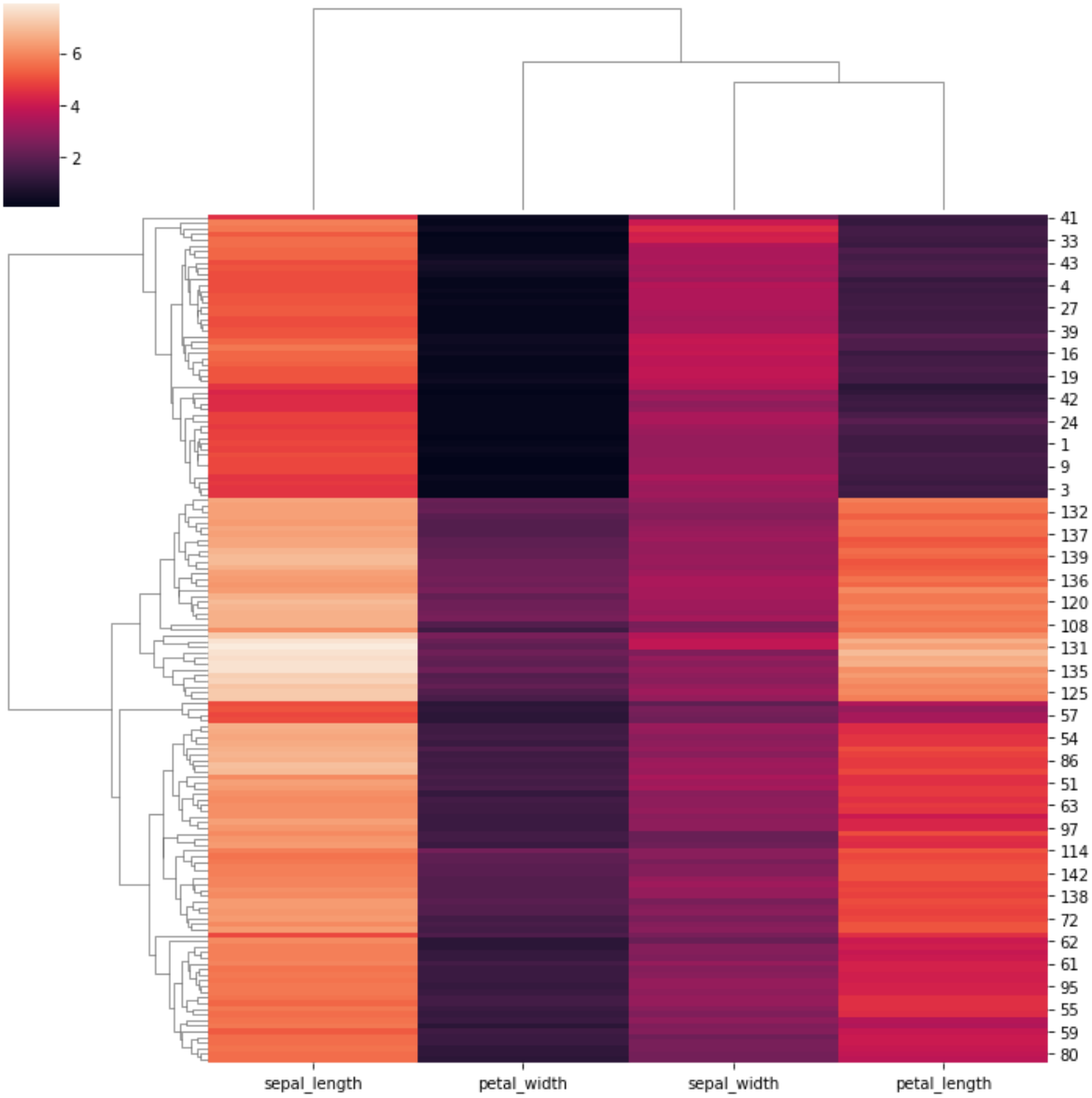
Out[77]:

	sepal_length	sepal_width	petal_length	petal_width	species	species_id
0	5.1	3.5	1.4	0.2	setosa	1
1	4.9	3.0	1.4	0.2	setosa	1
2	4.7	3.2	1.3	0.2	setosa	1
3	4.6	3.1	1.5	0.2	setosa	1
4	5.0	3.6	1.4	0.2	setosa	1
...	...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica	3
146	6.3	2.5	5.0	1.9	virginica	3
147	6.5	3.0	5.2	2.0	virginica	3
148	6.2	3.4	5.4	2.3	virginica	3
149	5.9	3.0	5.1	1.8	virginica	3

150 rows × 6 columns

In [78]:
```
sns.clustermap(iris.iloc[:,[0,1,2,3]])
```

Out[78]:
```
<seaborn.matrix.ClusterGrid at 0x1a0ebbab970>
```



In [79]:
```
import seaborn as sns
import plotly.express as px
```

Categorical Plots 1.Categorical scatter plot -stripplot -swarmplot

2.Categorical Distribution Plots -Boxplot -Violiniplot

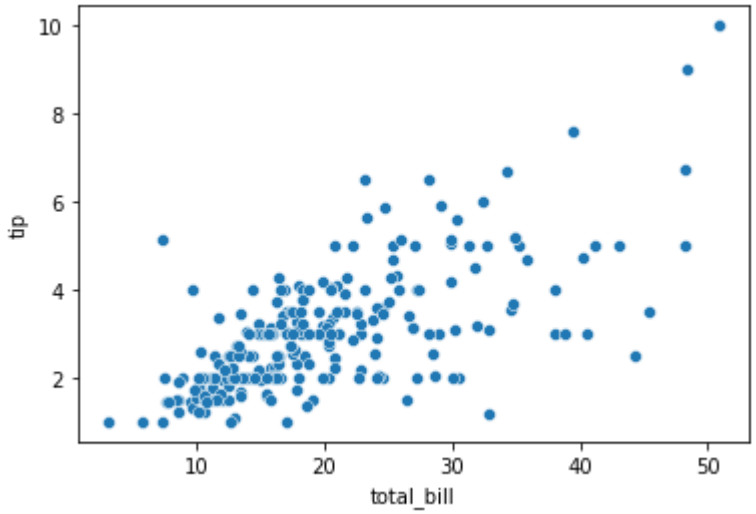
Categorical Estimate plot->for central -Barplot -pointplot -countplot

Figure level function->catplot

In [80]:
```
#import dataset
tips=sns.load_dataset("tips")
iris=sns.load_dataset("iris")
```

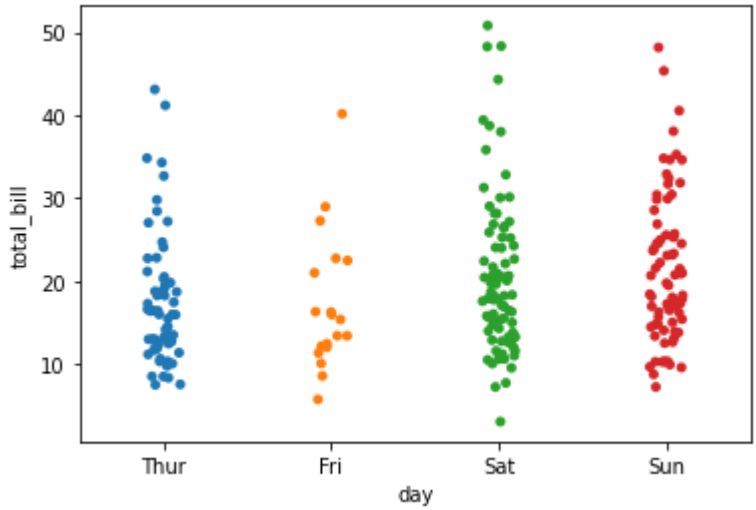
In [81]:
```
sns.scatterplot(data=tips,x="total_bill",y="tip")
```

Out[81]:
```
<AxesSubplot:xlabel='total_bill', ylabel='tip'>
```



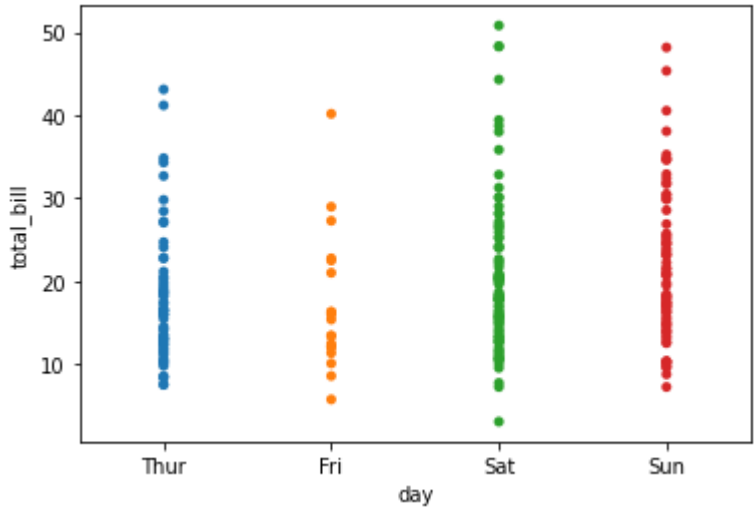
In [82]:
```
#strip plot
sns.stripplot(data=tips,x="day",y="total_bill")
```

Out[82]:
```
<AxesSubplot:xlabel='day', ylabel='total_bill'>
```



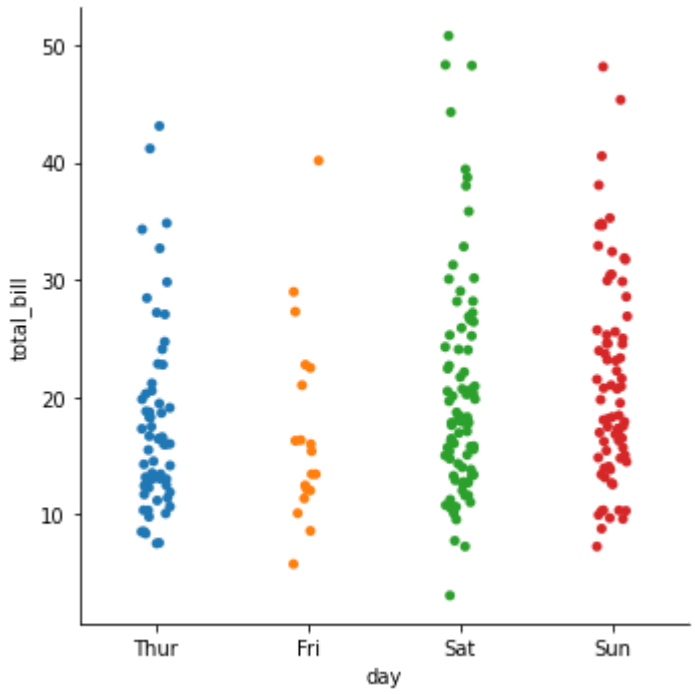
In [83]:
```
sns.stripplot(data=tips,x="day",y="total_bill",jitter=False)
```

Out[83]:
```
<AxesSubplot:xlabel='day', ylabel='total_bill'>
```



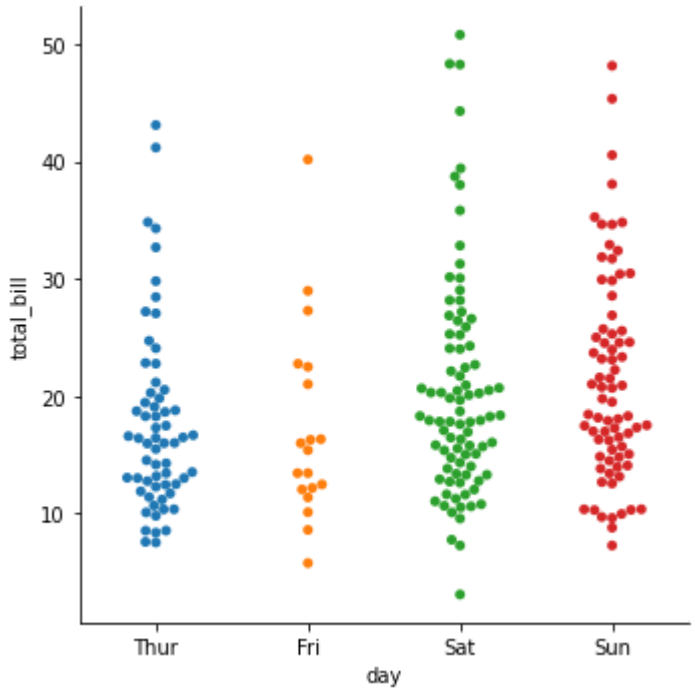
In [84]:
```
#using catplot
sns.catplot(data=tips,x="day",y="total_bill",kind="strip")
```

Out[84]: <seaborn.axisgrid.FacetGrid at 0x1a0ebe65dc0>



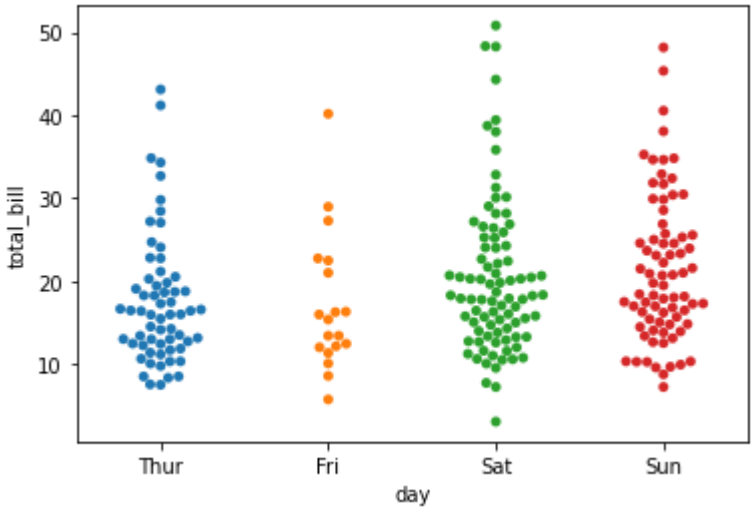
In [85]:
```
#swarmplot
sns.catplot(data=tips,x="day",y="total_bill",kind="swarm")
```

Out[85]: <seaborn.axisgrid.FacetGrid at 0x1a0ee4a8b80>



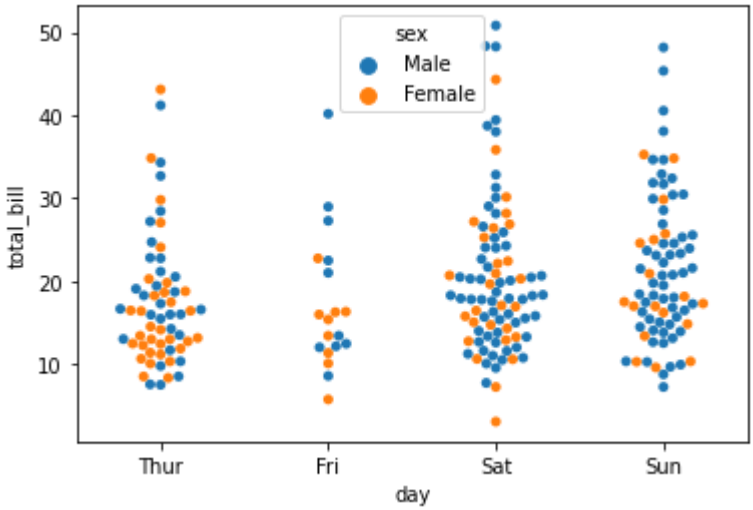
In [86]:
```
sns.swarmplot(data=tips,x="day",y="total_bill")
```

Out[86]: <AxesSubplot:xlabel='day', ylabel='total_bill'>



In [87]:
```
sns.swarmplot(data=tips,x="day",y="total_bill",hue="sex")
```

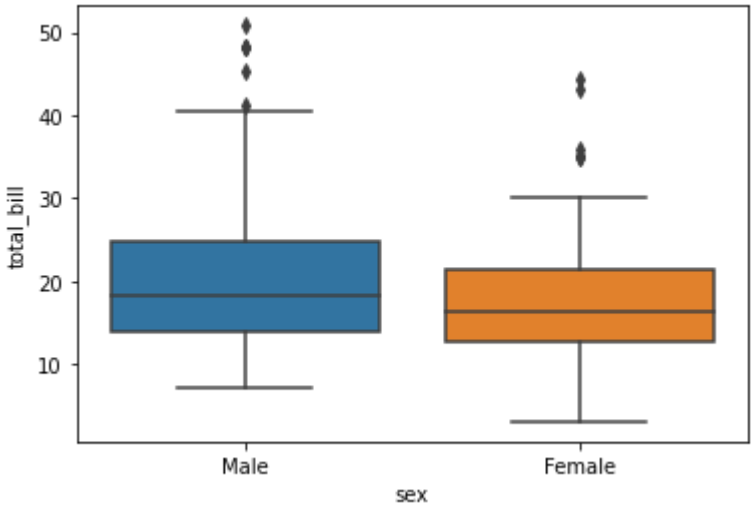
Out[87]:
```
<AxesSubplot:xlabel='day', ylabel='total_bill'>
```



Boxplot A boxplot is a standaradized way or displaying the distribution of data based on a five number summary (minimum,first quartile,median,third quartile and maximum)it also tell about data is symmetrical,how tightly data is grouped and if and how data is skewed.

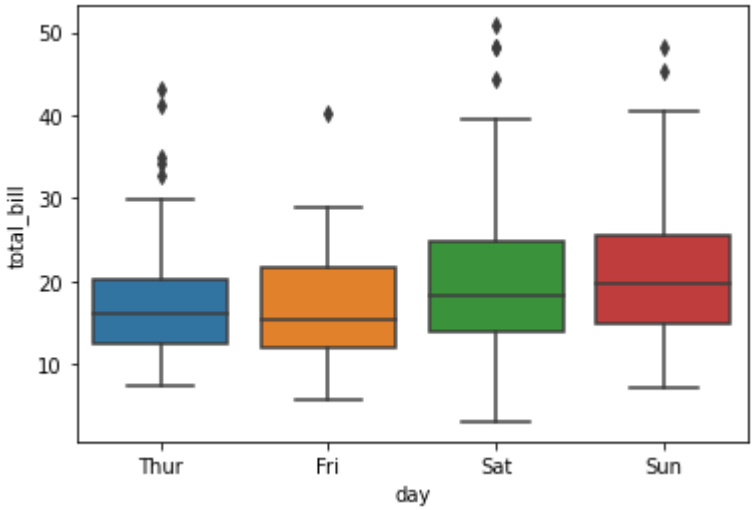
In [88]:
```
#Box Blot
sns.boxplot(data=tips,x="sex",y="total_bill")
```

Out[88]:
```
<AxesSubplot:xlabel='sex', ylabel='total_bill'>
```



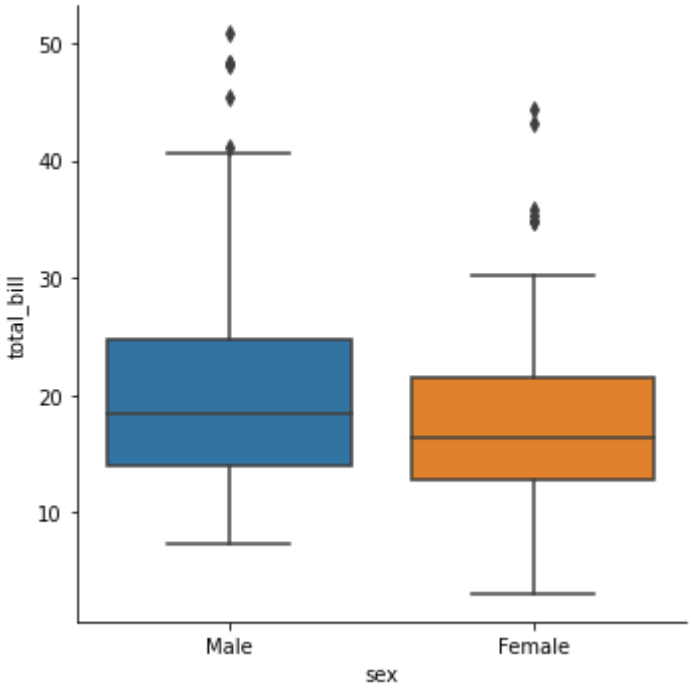
In [89]:
```
sns.boxplot(data=tips,x="day",y="total_bill")
```

Out[89]:
```
<AxesSubplot:xlabel='day', ylabel='total_bill'>
```



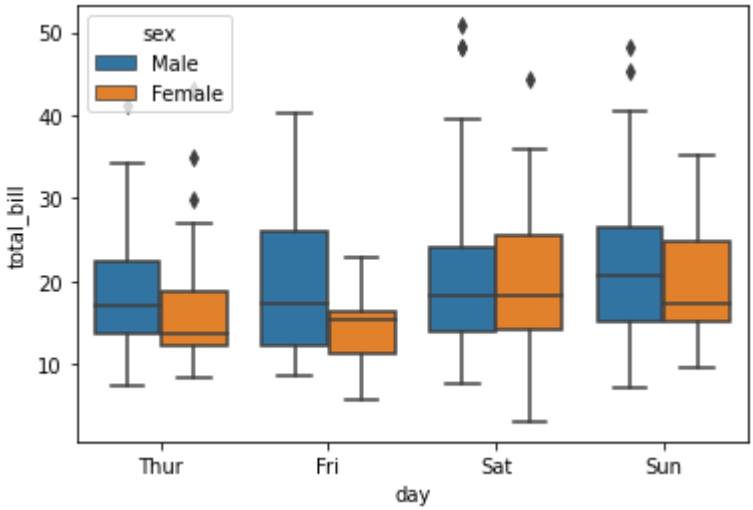
In [90]:
```
#catplot
sns.catplot(data=tips,x="sex",y="total_bill",kind="box")
```

Out[90]:
```
<seaborn.axisgrid.FacetGrid at 0x1a0e190fa90>
```



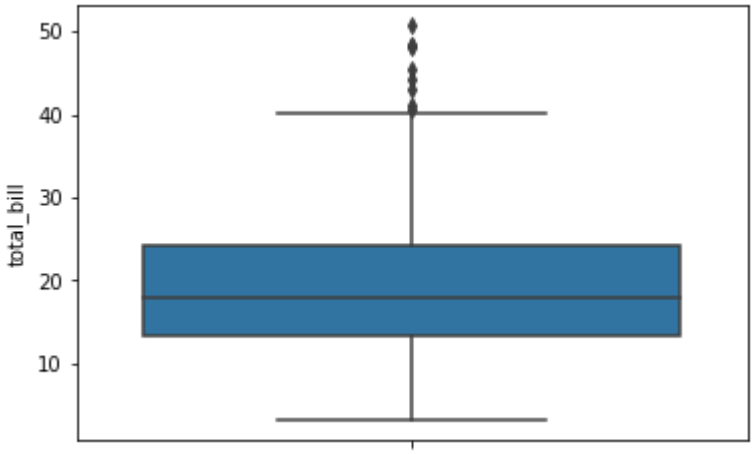
In [91]:
```
sns.boxplot(data=tips,x="day",y="total_bill",hue="sex")
```

Out[91]:
```
<AxesSubplot:xlabel='day', ylabel='total_bill'>
```



In [93]:
```
#single boxplot->numerical col
sns.boxplot(data=tips,y="total_bill")
```

Out[93]:
```
<AxesSubplot:ylabel='total_bill'>
```

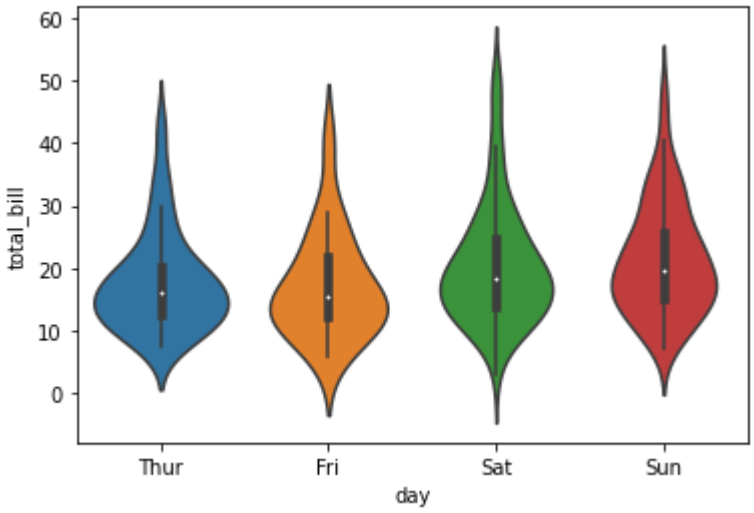


In [ ]:

Violinplot=(Boxplot+KDEplot)

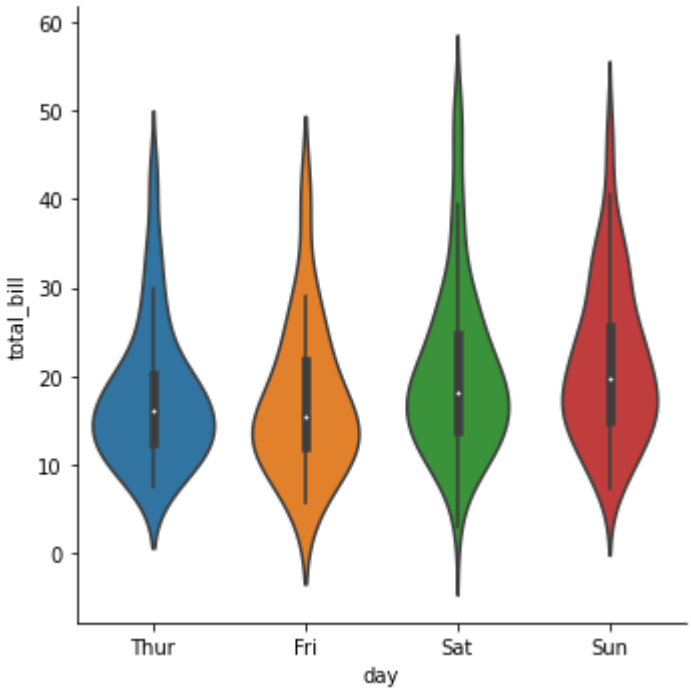
In [95]:
```
#violineplot
sns.violinplot(data=tips,x="day",y="total_bill")
```

Out[95]:
```
<AxesSubplot:xlabel='day', ylabel='total_bill'>
```



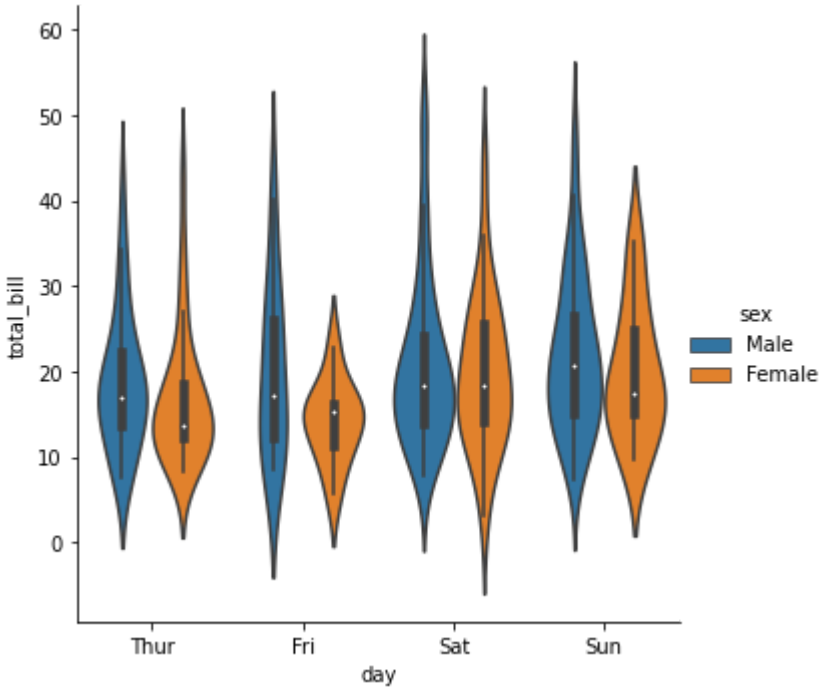
In [96]:
```
sns.catplot(data=tips,x="day",y="total_bill",kind="violin")
```

Out[96]:
```
<seaborn.axisgrid.FacetGrid at 0x1a0ef2f4cd0>
```



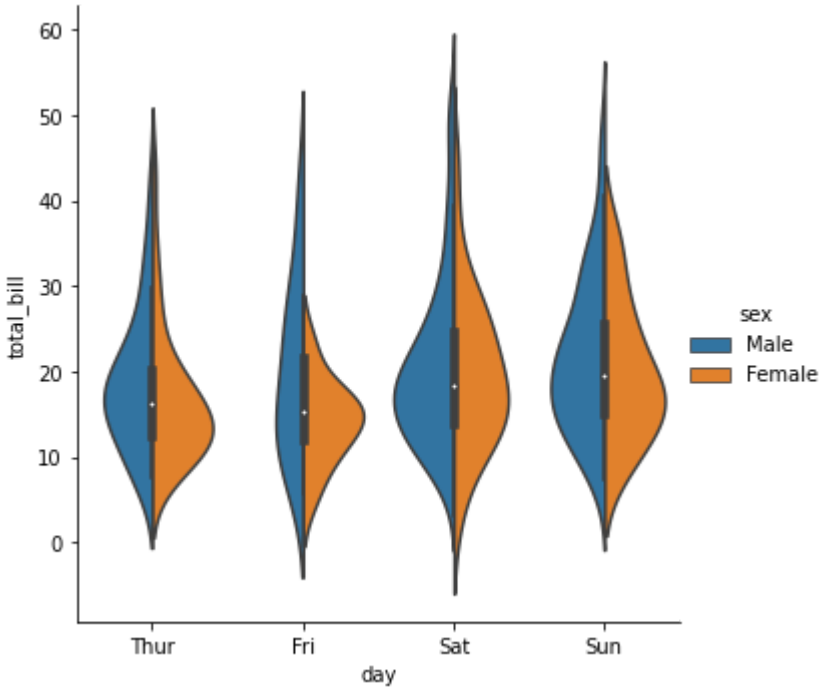
In [97]:
```
sns.catplot(data=tips,x="day",y="total_bill",kind="violin",hue="sex")
```

Out[97]:
```
<seaborn.axisgrid.FacetGrid at 0x1a0ef1ef910>
```



In [98]:
```
sns.catplot(data=tips,x="day",y="total_bill",kind="violin",hue="sex",split=True)
```

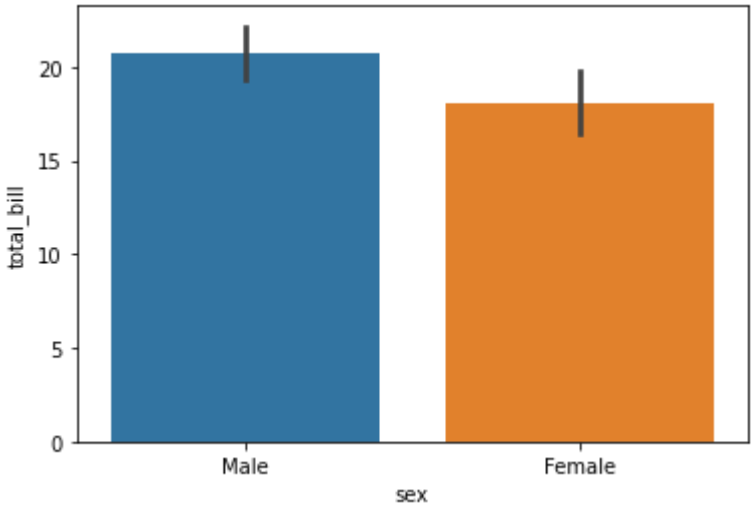
Out[98]:
```
<seaborn.axisgrid.FacetGrid at 0x1a0f0e1f460>
```



Barplot When there are multiple observation in each category,it also uses bootstapping to compute a confidence interval around the estimate,which is plotted using error bars.

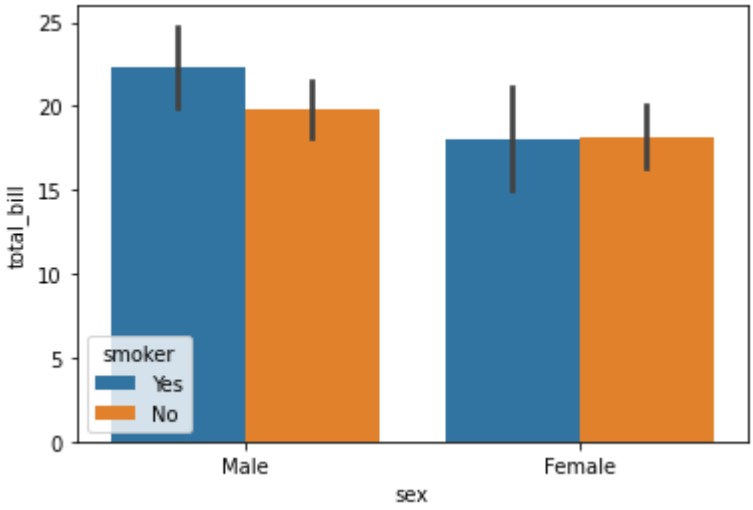
In [99]:
```
sns.barplot(data=tips,x="sex",y="total_bill")
```

Out[99]:
```
<AxesSubplot:xlabel='sex', ylabel='total_bill'>
```



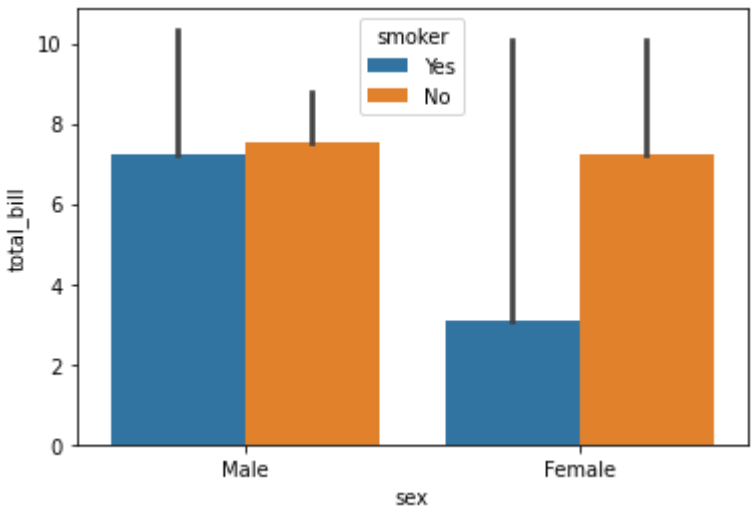
In [100…
```
sns.barplot(data=tips,x="sex",y="total_bill",hue="smoker")
```

Out[100]:
```
<AxesSubplot:xlabel='sex', ylabel='total_bill'>
```



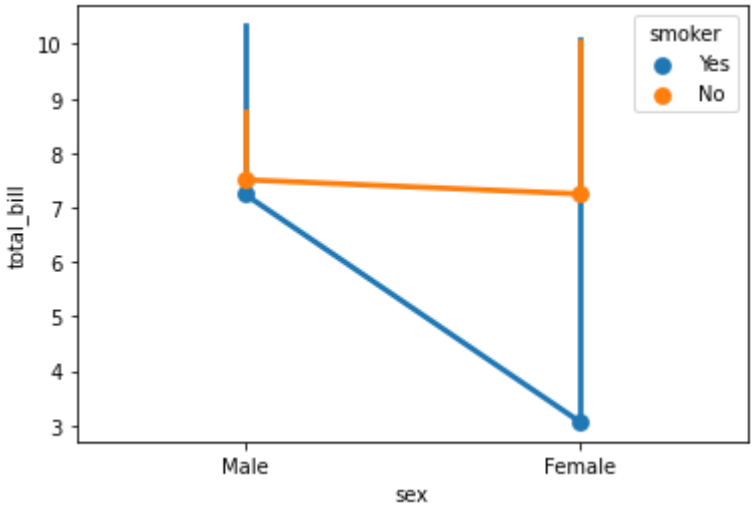
In [101…
```
import numpy as np
sns.barplot(data=tips,x="sex",y="total_bill",hue="smoker",estimator=np.min)
```

Out[101]:
```
<AxesSubplot:xlabel='sex', ylabel='total_bill'>
```



In [102…
```
sns.pointplot(data=tips,x="sex",y="total_bill",hue="smoker",estimator=np.min)
```

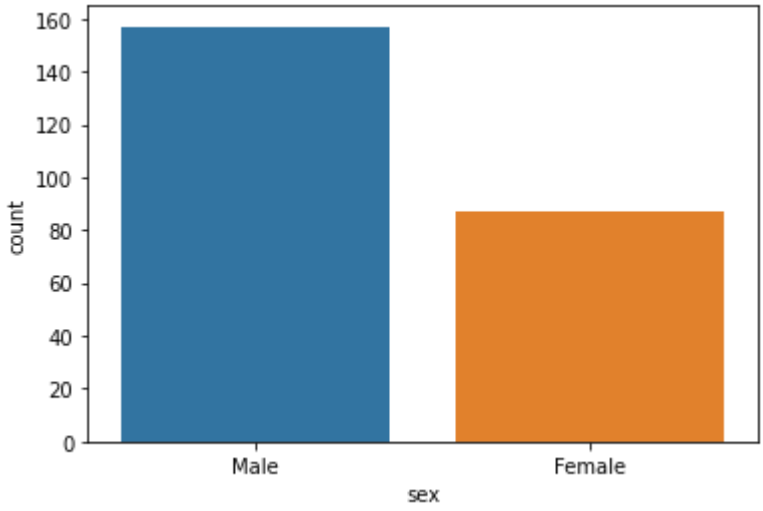
Out[102]:
```
<AxesSubplot:xlabel='sex', ylabel='total_bill'>
```



Count Plot a special case for the bar when we want to the number of observation in each category rather than computing a statistic for a second variable.This is similar to a histogram over a categorical rathar than quantities variable

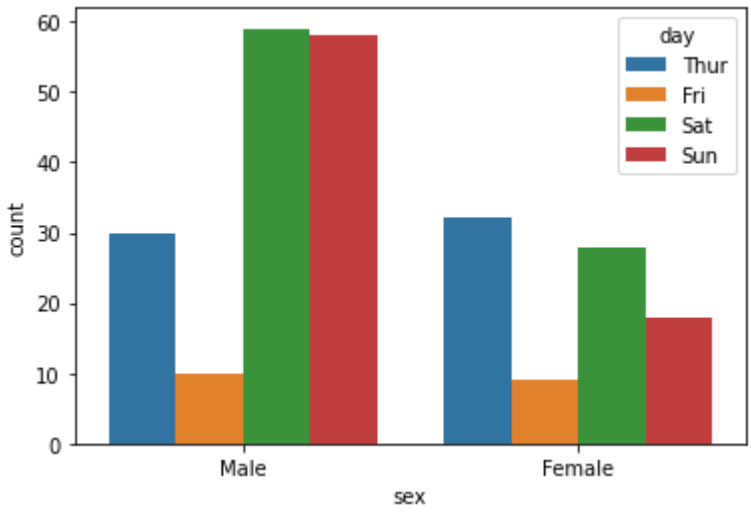
In [103…
```
sns.countplot(data=tips,x="sex")
```

Out[103]:
```
<AxesSubplot:xlabel='sex', ylabel='count'>
```



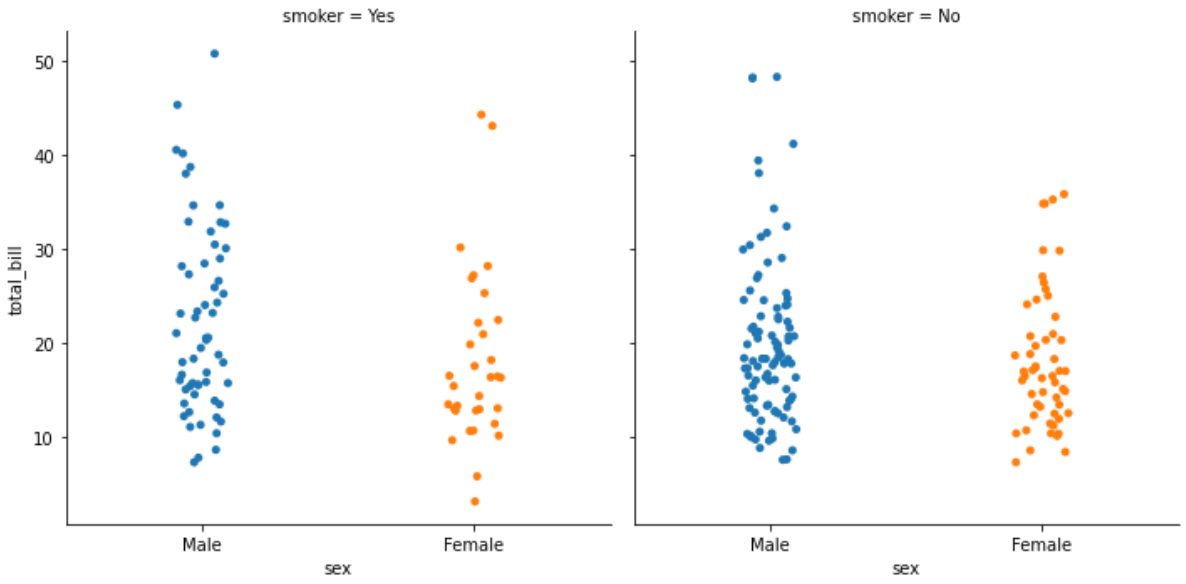
In [104…
```
sns.countplot(data=tips,x="sex",hue="day")
```

Out[104]:
```
<AxesSubplot:xlabel='sex', ylabel='count'>
```



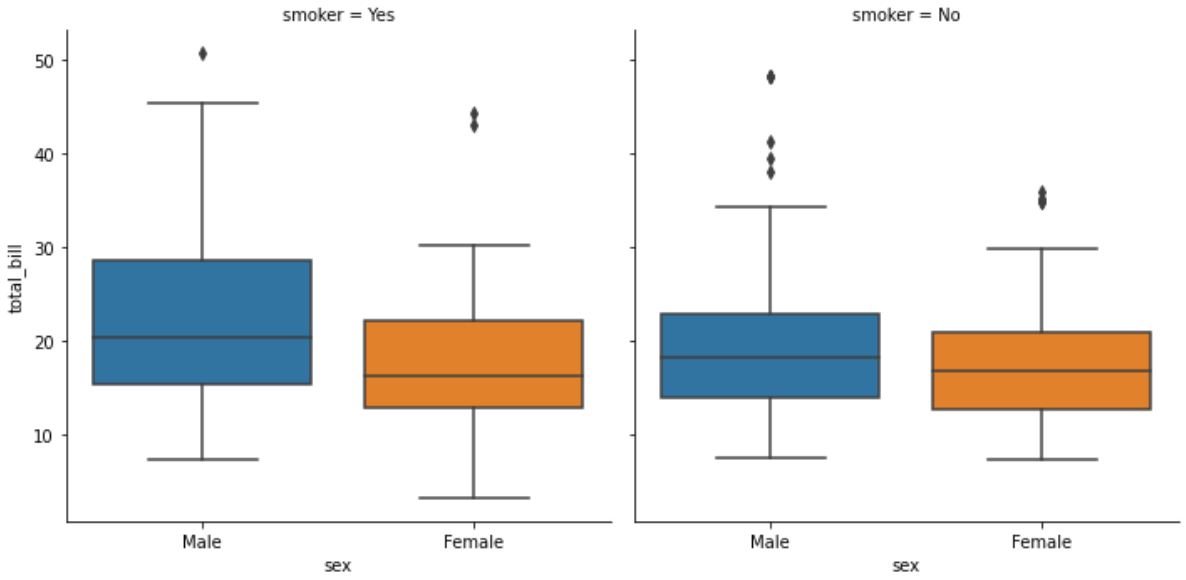
In [105…
```
#faceting using catplot
sns.catplot(data=tips,x="sex",y="total_bill",col="smoker")
```

Out[105]:
```
<seaborn.axisgrid.FacetGrid at 0x1a0f0d456d0>
```



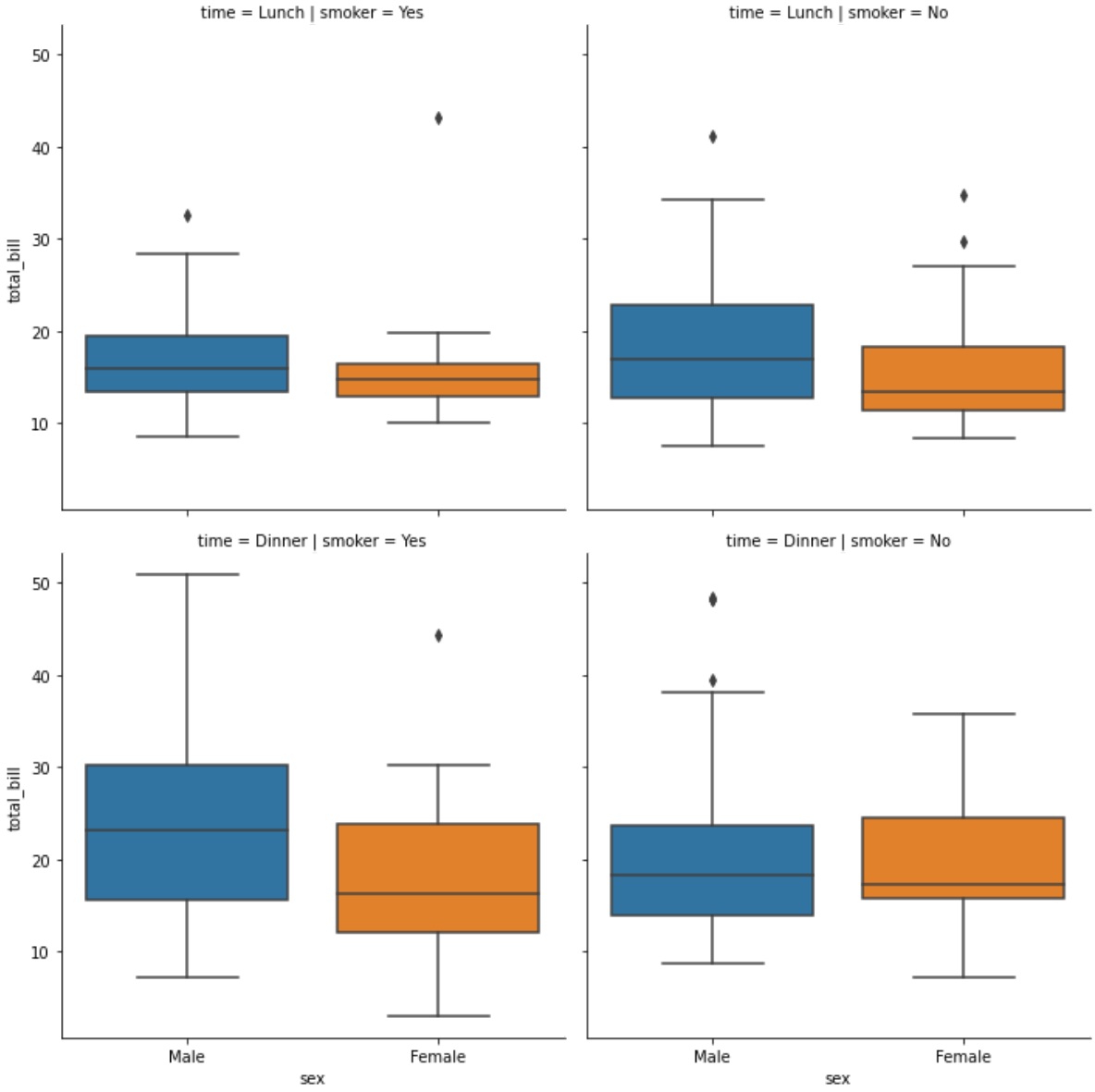
In [106…
```
sns.catplot(data=tips,x="sex",y="total_bill",col="smoker",kind="box")
```

Out[106]:
```
<seaborn.axisgrid.FacetGrid at 0x1a0f1008280>
```



In [107…
```
sns.catplot(data=tips,x="sex",y="total_bill",col="smoker",kind="box",row="time")
```

Out[107]:
```
<seaborn.axisgrid.FacetGrid at 0x1a0f1261820>
```

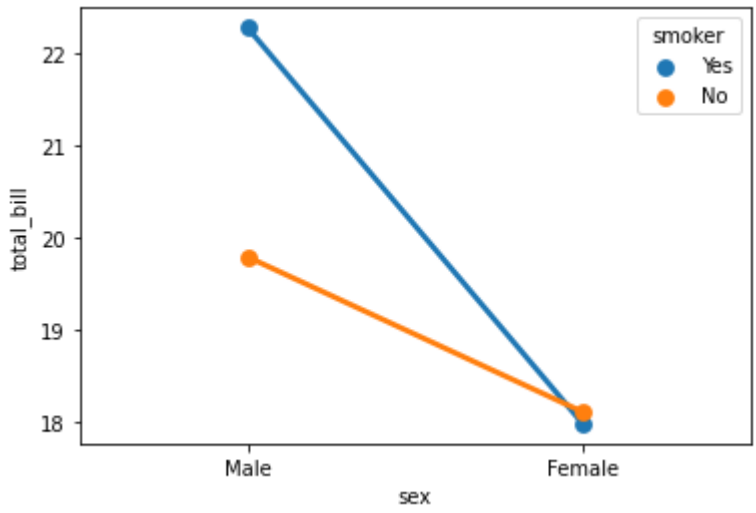


In [108…

```
#point plot
sns.pointplot(data=tips,x="sex",y="total_bill",hue="smoker",ci=None)
```

Out[108]:

```
<AxesSubplot:xlabel='sex', ylabel='total_bill'>
```



In [ ]: